DISK TESTS RK05

1. RK11 DISK DATA
2. RK11 RANDOM EXERCISER
3. PDP-11 DEVICE DRIVER PACKAGE

digital

Customer: Digitgraph

Attention:

SUBJECT: PRELIMINARY MATERIAL

From: Waiver Control Group
Bldg. 1 Floor 1
Maynard, Ma 01754

DEC Order No.: E8999

System No.: PDP-11/~~719~~40XXX System No. 719

The enclosed material replaces preliminary items shipped with the above order as shown on return reply card.

☐ Order Complete ☒ Incomplete

1 - Maindec-11-D5HA PB 1 & 2 & D
Replaces Maindec-11-DZRKA & DZRKB PB & D

DIGITAL EQUIPMENT CORPORATION, 146 MAIN STREET, MAYNARD, MASSACHUSETTS 01754
(617)897-5111 TWX: 710-347-0212 TELEX: 94-8457

IDENTIFICATION

PRODUCT CODE	MAINDEC-11-DZRKA-A-D (REPLACES: MAINDEC-11-D5HA-D)
PRODUCT NAME	RK11 STATIC TEST
DATE CREATED	MARCH 5, 1973
MAINTAINER	DIAGNOSTIC GROUP
AUTHOR	E.HAIGHT, C. CASWELL

COPYRIGHT © 1971, 1973
DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

SECTION	CONTENTS
1,	ABSTRACT
2,	REQUIREMENTS
2,1	EQUIPMENT
2,2	STORAGE
3,	LOADING PROCEDURE
3,1	METHOD
4,	STARTING PROCEDURE
4,1	WORST CASE OPERATION
4,2	ADDRESS ENTRY POINTS
4,3	MAINTENANCE ROUTINES
5,	OPERATING PROCEDURE
5,1	CONTROL SWITCH SETTINGS
5,2	SUBROUTINE ABSTRACT
6,	ERROR REPORTS
7,0	MISCELLANEOUS
8,	RUNTIME

1. ABSTRACT

THE RK11 DISK DATA TEST IS A SERIES OF ADDRESS AND DATA RELIABILITY ROUTINES WHICH VERIFY TO THE USER THE DISK CONTROL (RK11) AND DISK (RK02 THRU RK05) ARE OPERATING CORRECTLY. THIS TEST USED IN CONJUNCTION WITH THE RK11 DATA TEST AND RK11 RANDOM DRIVE EXERCISER ASSURES THE USER OF AN ERROR FREE SYSTEM, WHEN USED IN ITS ENTIRETY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11
RK11 AND (RK02 THRU RK05)

2.2 STORAGE
PROGRAM OCCUPIES FROM 0 TO 11574

2.3 PRELIMINARY PROGRAMS
PROCESSOR AND MEMORY TESTS

3. LOADING PROCEDURE

3.1 METHOD OF LOADING DISK DATA TEST TAPE

PROGRAM FORMAT ABSOLUTE

A. VERIFY THE BOOT LOADER IS IN MEMORY.

B. SET SWITCH REGISTER EQUAL TO #750

MEMORY SIZE	#
4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

C. DEPRESS LOAD ADDRESS.

D. DEPRESS START.

4. STARTING PROCEDURE

4.1 WORST CASE TEST DRIVE ZERO FOR STATIC TEST

A) TURN ALL DRIVES EXCEPT DRIVE 0 OFF LINE,
B) SET SWITCH REGISTER EQUAL TO 200
C) DEPRESS LOAD ADDRESS
D) SET SWITCH REGISTER EQUAL TO ZERO
E) DEPRESS START

4.2 ADDRESS ENTRY POINTS

300	STA11	TEST RESET FUNCTION
304	STA12	TEST RESET CURRENT ADDRESS REGISTER
310	STA13	TEST RESET WORD COUNT REGISTER
314	STA14	TEST RESET DRIVE ADDRESS REGISTER
320	STA15	TEST RESET CONTROL ERROR REGISTER
324	STA16	TEST RESET DATA BUFFER REGISTER
330	STA17	TEST RESET MAINTENANCE REGISTER
334	STA110	TEST CAN WE SET FUNCTION REGISTER BITS
340	STA111	TEST CAN WE CLEAR FUNCTION REGISTER BITS
344	STA112	TEST CAN WE SET ALL CMA BITS
350	STA113	TEST CAN WE CLEAR ALL CMA BITS
354	STA114	TEST CAN WE SET ALL WORD COUNT BITS
360	STA115	TEST CAN WE CLEAR ALL WORD COUNT BITS
364	STA116	TEST CAN WE SET ALL DRIVE ADDRESS BITS
370	STA117	TEST CAN WE CLEAR ALL DRIVE ADRESS BITS
374	STA122	EXECUTE A ONE WORD WRITE
400	STA140	EXECUTE A ONE WORD WRITE CHECK
404	STA156	EXECUTE A ONE WORD READ
410	INHCMA	TEST CURRENT ADDRESS INHIBIT
414	STA174	TEST THE SURFACE INCREMENT
420	STA177	TEST THE CYLINDER INCREMENT

(4.2 CONT'D)

424	STA103	TEST THE OVERRUN ERROR
430	NXDST	TEST THAT NED RAISES ERROR FLAG
434	ST103X	TEST THE NON-EXISTENT SECTOR ERROR
440	ST104X	TEST THE NON-EXISTENT CYLINDER ERROR
444	ST105X	TEST THE PROGRAM ERROR
450	NXMTSM	TEST THE NON-EXISTENT MEMORY ERROR
454	STA106	TEST THE CONTROL WILL NOT TRAP AT BR7
460	STA110	TEST THE CONTROL WILL NOT TRAP AT BR6
464	STA112	TEST THE CONTROL WILL NOT TRAP AT BR5
470	STA114	TEST THE CONTROL WILL TRAP AT BR4
474	RESTORE	TEST DRIVE RESTORE
500	ADT1	TEST THE DRIVE SECTOR FORMAT

4.3 ***MAINTENANCE ROUTINES*** (BOTH STATIC AND DATA TESTS)

634	SELWC	LOAD WORD COUNT REG. WITH SWITCH REGISTER
640	SELCMA	LOAD CURRENT MEMORY ADDR REG. WITH SWITCH REGISTER
644	SELDAR	LOAD DISK ADDR. REGISTER WITH SWITCH REGISTER
650	SELDCS	LOAD DISK CONTROL REGISTER WITH SWITCH REGISTER
654	TKSEL	LOAD CYLINDER ADDRESS REGISTER FROM SWITCH REGISTER AND SEEK.

5. OPERATING PROCEDURE

5.1 CONTROL SWITCH SETTINGS

DELETE TYPEOUT

SR14 SET DELETE TYPEOUTS

RESET REPORT MESSAGE

(5.1 CONT'D)

LOOP ON TEST

SR11 SET LOOP ON TEST

RESET CONTINUE TO NEXT TEST

HALT ON ERROR

SR10 SET HALT AFTER ERROR REPORT

RESET CONTINUE AFTER ERROR REPORT

SR7-0 SELECT CYLINDER FROM SR

CYLINDER SELECTION

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

SELECT ONE OF 312(8) CYLINDERS

5.2 SUBROUTINE ABSTRACTS

THE PROGRAM CONSIST OF MANY SMALL TESTS EACH OF WHICH HAS BREIF DESCRIPTION AT THE BEGINNING.

TKSEL = OPERATING PROCEDURE

STEP A: SET SWITCH REGISTER EQUAL TO 654

STEP B: DEPRESS LOAD ADDRESS

STEP C: SET SWITCH REGISTER 10 THRU 8 EQUAL TO DRIVE #0 THRU 7

(5.2 CONT'D)

STEP D, DEPRESS START

STEP E, SET SWITCH REGISTER 7 THRU 0 EQUAL TO CYLINDER #

STEP F, DEPRESS CONTINUE

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

CYLINDER NUMBER
(0 THRU 177(8))

STEP G, TRACK NUMBER CAN BE CHANGED ARBITRARILY.

6. ERROR REPORTS

6.1 STATIC AND ADDRESS REPORT,
XX ERR CNT XXXXXXWRD1XXXXWRD2

ERRCNT = IS THE TAG FOR THE LISTING
WRD1 = WHAT WAS EXPECTED
WRD2 = WHAT WAS RECEIVED

WHEN A REPORT ONLY CONTAINS ONE WORD THE PROGRAM WAS EXPECTING ZEROS BUT RECEIVED WHAT WAS REPORTED.

6.2 ERROR REPORTS

STATUS ERROR

STATUS ERROR XXXXXXRKDA XXXXXXRKER XXXXXXRKCS

A B C

A=THE DRIVE ADDRESS REGISTER
B=THE CONTROL ERROR REGISTER
C=THE CONTROL FUNCTION REGISTER

BIT LAYOUT OF RKCS REGISTER
BIT15= ERROR
BIT14= HARD ERROR
BIT13= SEARCH COMPLETE
BIT12= MAINTENANCE ENABLE
BIT11= NOT USED
BIT10= FORMAT
BIT9 = READ/WRITE ALL
BIT8 = STOP ON SOFT ERROR
BIT7= READY
BIT6= INTERRUPT ENABLE
BIT5= EXTENDED MEMORY 1 (XM1)
BIT4= EXTENDED MEMORY 0 (XM0)
BIT3-2-1= FUNCTION REGISTER
BIT0= GO

(6,2 CONT'D)

BIT 3	BIT 2	BIT 1	OPERATION
0	0	0	RESET
0	1	0	READ
0	0	1	WRITE
0	1	1	WRITE CHECK
1	0	0	SEEK
1	0	1	READ CHECK
1	1	0	DRIVE RESET
1	1	1	WRITE LOCK

BIT LAYOUT OF RKER REGISTER

BIT15= DRIVE ERROR (DRE)
BIT14= OVERRUN (OVR)
BIT13= WRITE LOCK OUT VIOLATION (WLO)
BIT12= SEEK ERROR (SKE)
BIT11= PROGRAMMING ERROR (PCE)
BIT10= NON-EXISTENT MEMORY ERROR (NXM)
BIT9 = DATA LATE ERROR (DTL)
BIT8 = READ TIMING ERROR (RTE)
BIT7 = NON-EXISTENT DISK ERROR (NXD)
BIT6 = NON-EXISTENT CYLINDER ERROR (NXC)
BIT5 = NON-EXISTENT SECTOR (NXS)
BIT4-3-2= NOT USED
BIT1 = CHECKSUM ERROR (CSE)
BIT0 = WRITE CHECK ERROR (WCE)

IF THE ERROR OCCURRED WHEN READING THE PROGRAM WILL REPORT WHICH READ.

NOTE: WHEN A HARD ERROR OCCURS AN ADDITIONAL ERROR MESSAGE WILL BE REPORTED, AS FOLLOWS:

XXXXXX RKDS

LAYOUT OF RKDS BITS

BIT15-14-13= IDENTIFICATION OF DRIVE
BIT12= DRIVE POWER LOW (DPL)
BIT11= HIGH DENSITY DRIVE (HDEN)
BIT10= DRIVE UNSAFE (DRU)
BIT9 = SEEK INCOMPLETE (SIN)
BIT8 = SECTOR COUNTER OK (SOK)
BIT7 = DRIVE READY (DRY)
BIT6 = ACCESS READY (ARDY)
BIT5 = WRITE PROTECT STATUS (WPS)
BIT4 = SECTOR COUNTER EQUALS SECTOR ADDRESS (SC=SA)
BIT3-0= SECTOR COUNTER (S.C.)

6,3 PARITY ERROR

THIS MESSAGE IS REPORTED IF THE PROGRAM DETECTS A PARITY ERROR DURING PROGRAM EXECUTION,

6,4 END

END

THIS MESSAGE IS REPORTED AT THE END OF ONE COMPLETE PASS OF THE DISK SYSTEM,

7, MISCELLANEOUS

IN SOME ADDRESS TESTS THE PROGRAM DEPENS ON WRITTING AND READING DATA CORRECTLY FROM THE DISK,AND IF IT DOES NOT IT MAY REPORT AN ADDRESS FAILURE, WHEN IN FACT IT WAS A DATA FAILURE;

8, RUNTIME

TYPEOUT WILL OCCUR WITHIN 5 MIN,

```
                                ;TITLE MAINDEC-11-DZRKA-A RK11 STATIC TEST REPLACES DSHA
                                ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
                                ;PROGRAM BY E. HAIGHT/C. CASWELL
                                .ENABL  ABS

            000000              N=0
            000001              BI↑0=1
            000002              BI↑1=2
            000004              BI↑2=4
            000010              BI↑3=10
            000020              BI↑4=20
            000040              BI↑5=40
            000100              BI↑6=100
            000200              BI↑7=200
            000400              BI↑8=400
            001000              BI↑9=1000
            002000              BI↑10=2000
            004000              BI↑11=4000
            010000              BI↑12=10000
            020000              BI↑13=20000
            040000              BI↑14=40000
            100000              BI↑15=100000
            000004              SCOPE=!OT
                                ;
                                ;
                                ;
            104403              WRITE=TRAP+3
            104407              WRCHECK=TRAP+7
            104405              READ=TRAP+5
            000000              .=0
                                ;TRAP CATCHER 0 -=> 776
                                ;MACRO FOR SETTING UP ERROR COUNT
            000200              .=200
000200  000167  000676          JMP     START
                                ;STATIC ROUTINES
            000300                      .=300
000300  000167  000726                  JMP     STAI1           ;TEST RESET TO CONTROL REGISTER
000304  000167  001000                  JMP     STAI2           ;TEST RESET TO CURRENT ADDRESS REGISTER
000310  000167  001024                  JMP     STAI3           ;TEST RESET TO WORD COUNT REGISTER
000314  000167  001050                  JMP     STAI4           ;TEST RESET TO DISK ADDRESS REGISTER
000320  000167  001074                  JMP     STAI5           ;TEST RESET TO CONTROL ERROR REGISTER
000324  000167  001120                  JMP     STAI6           ;TEST RESET TO DATE BUFFER REGISTER
000330  000167  001144                  JMP     STAI7           ;TEST RESET TO MAINTENANCE REGISTER
000334  000167  001170                  JMP     STAI10          ;CAN WE SET W/R BITS IN DCS REGISTER
000340  000167  001242                  JMP     STAI11          ;CAN WE CLEAR THE DCS REGISTER
                                                                ;USING DISK CLEAR,
000344  000167  001322                  JMP     STAI12          ;CAN WE SET ALL CMA BITS
000350  000167  001366                  JMP     STAI13          ;CAN WE CLEAR ALL CMA BITS
                                                                ;USING DISK CLEAR
000354  000167  001436                  JMP     STAI14          ;CAN WE SET ALL WC BITS
```

```
    55  000360  000167  001502                  JMP     STA15           ;CAN WE CLEAR ALL WC BITS
    56                                                                    ;USING DISK CLEAR
    57  000364  000167  001546                  JMP     STA16           ;CAN WE SET ALL THE RKDA BITS
    58  000370  000167  001612                  JMP     STA17           ;CAN WE CLEAR ALL THE DAR BITS
    59                                                                    ;USING DISK CLEAR
```

```
                                                ;EXECUTE A ONE WORD WRITE
                                                ;FOLLOWED BY A ONE WORD WRITE CONTINUE
                                                ;EXECUTE A ONE WORD WRITE CHECK
                                                ;FOLLOWED BY A ONE WORD WRITE CHECK CONTINUE
                                                ;EXECUTE A ONE WORD READ
                                                ;FOLLOWED BY A ONE WORD READ CONTINUE
                                                ;TEST CURRENT ADDRESS INHIBT
                                                ;TEST SURFACE INCREMENT
                                                ;TEST CYLINDER INCREMENT
                                                ;TEST OVERRUN ERROR
                                                ;TEST THAT NED RAISES ERROR FLAG
                                                ;TEST NON-EXISTENT SECTOR
                                                ;TEST NON-EXISTENT CYLINDER
                                                ;CHECK PROGRAM ERROR
                                                ;TEST NON-EXISTENT MEMORY ERROR
                                                ;TEST THAT THE DISK WILL NOT TRAP
                                                ;AT PRIORITY 7
                                                ;TEST THAT THE DISK WILL NOT TRAP
                                                ;AT PRIORITY 6
                                                ;TEST THAT THE DISK WILL NOT TRAP
                                                ;AT PRIORITY 5
                                                ;TEAST THAT THE DISK WILL TRAP
                                                ;AT PRIORITY 4
                                                ;TEST RESTORE FUNCTION
                                                ;CHECK SECTOR FORMAT
                    .=634
                    ;
                    ;MAINTENANCE ROUTINES
                    ;
                                                ;LOAD WORD COUNT REG WITH SR
                                                ;LOAD CURRENT ADDRESS REG WITH SR
                                                ;LOAD DISK ADDRESS REG WITH SR
                                                ;LOAD FUNCTION REG WITH SR
                    ;
                    ;
                                                ;SELECT CYLINDERS STATICLY
                    ;
                    ;
                    ;
                    ;
                    ;
                    ;
                    ;RK11 DATA TEST
                    ;VECTORS USED IN PROGRAM
                    ;#1 LOC 220 DISK INTERRUPT
                    ;#2 LOC 30 EMT (TELETYPE OUTPUT)
                    ;#3 LOC 34 TRAP (DISK HANDLERS)
                    ;
                    .=1000
```

```
                                        .EVEN
                                        ;
                                        ;I/O ADDRESS POINTERS
001000  177570                          SWR:    177570                  ;SWITCH REGISTER
001002  177776                          PS:     177776                  ;PROCESSOR STATUS REGISTER
001004  177566                          TPB:    177566                  ;TELETYPE REGISTERS
001006  177562                          TKB:    177562
001010  177564                          TPS:    177564
001012  177560                          TKS:    177560
                                        ;
                                        ;DISK I/O REGISTERS
                                        ;
001014  177404                          RKCS:   177404                  ;DISK CONTROL REGISTER
001016  177406                          RKWC:   177406                  ;WORD COUNT REGISTER
001020  177410                          RKBA:   177410                  ;CURRENT MEMORY ADDRESS REGISTER
001022  177412                          RKDA:   177412                  ;LOWER 16 BITS OF DISK ADDRESS
001024  177402                          RKER:   177402                  ;CONTROL ERROR REGISTER
001026  177400                          RKDS:   177400                  ;DRIVE STATUS REGISTER
001030  177416                          RKDB:   177416                  ;DATA BUFFER REGISTER
001032  177414                          RKMR:   177414                  ;RKMAINTENANCE REGISTER
001034  000220                          VECTOR: 220
001036  000222                          STATUS: 222
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;RK11 DEDICATE REGISTERS (MEMORY)
                                        ;
001040  000000                          FLAG:   0                       ;INTERNAL PROGRAM FLAG
001042  146723                          RANNU:  146723                  ;RANDOM NUMBER PRIME
001044  000000                          WRDCT:  0                       ;WRDING WORD COUNT
001046  000000                          TRACK:  0                       ;WRDING DAE
001050  000000                          DMA:    0                       ;WRDING RKDA
001052  000000                          PATNU:  0                       ;DATA PATTERN INDEX
001054  000000                          BUF:    0                       ;WRDING DATA BUFFER (OUT-IN)
001056  000000                          TWRDCT: 0                       ;TEMP WORD COUNT
001060  000000                          TDMA:   0                       ;TEMP RKDA
001062  000000                          SRDCT:  0                       ;STANDARD WORD COUNT
001064  000000                          ERCOUNT:0                       ;ERROR COUNT FOR MESSAGES,
001066  000000                          SAVE:   0
001070  000000                          SAV1:   0
001072  000000                          PASS:   0
                                        ;
                                        ;RF11 WRD REGISTERS
                                        ;(CAN BE CHANGED IN ANY ROUTINE)
001074  000000                          WRD:    0
001076  000000                          WRD1:   0
001100  000000                          WRD2:   0
```

```
                                                ;
001102  000005                  START:  RESET                   ;CLEAR THE WORLD
001104  012706  001000                  MOV     #1000,X6        ;SET UP STACK
001110  012767  000006  170666          MOV     #6,4
001116  005067  176664                  CLR     6
001122  012767  000340  177056          MOV     #340,2@6        ;LOCK UP INTERRUPTS
001130  012767  010676  170672          MOV     #EMTRP,3@       ;SET UP TTY POINTER
001136  012767  000340  170666          MOV     #340,32         ;LOCK UP INTERRUPTS
001144  012767  010150  170662          MOV     #DISK,34        ;SET UP DISK HANDLER POINTER
001152  012767  000340  170656          MOV     #340,36         ;LOCK UP INTERRRUPTS
001160  012777  000340  177614          MOV     #340,@PS        ;LOCK UP INTERRUPT LEVELS
001166  012767  010100  170624          MOV     #LOOP,2@        ;SET UP FOR SCOPE LOOP
001174  012767  000340  170620          MOV     #340,22         ;LOCK UP PRIORITY
001202  012767  010222  170614          MOV     #DOWN,24        ;SET UP POWER FAIL
001210  012767  000340  170610          MOV     #340,26         ;LOCK UP PRIORITIES
001216  005067  177624                  CLR     TRACK           ;CLEAR TRACK REGISTERS
001222  005067  177622                  CLR     DMA             ;CLEAR RKDA REGISTERS
001226  004767  007776                  JSR     X7,MAMF         ;SET UP PARITY SWITCHES
                                        ;WE ARE NOW ENTERING THE STATIC TEST
                                        ;IF THE OPERATOR WOULD LIKE TO CHECK
                                        ;THE DISK REGISTERS PRIOR TO ENTERING THIS
                                        ;TEST WE HAVE SOME HANDY ROUTINES
                                        ;WHICH WOULD ALLOW YOU TO LOAD THESE
                                        ;REGISTERS UNDER SWITCH REGISTER CONTROL
                                        ;PLEASE REFERENCE THE STARTING ADDRESS
                                        ;TO SEE WHICH ROUTINE BEST SUITS YOUR
                                        ;PROBLEM.
                                        ;
                                        ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
                                        ;TO CLEAR ALL THE DISK REGISTERS
                                        ;TEST CONTROL REGISTEN
001232  000005                  STA11:  RESET                   ;CLEAR THE WORLD
001234  017767  177554  177632          MOV     @RKCS,WRD       ;FETCH CONTROL REGISTER
001242  032767  177577  177624          BIT     #177577,WRD     ;IS IT CLEARED
001250  001405                          BEQ     XSTA11          ;REGISTER OK (TEST READY)
001252  012767  000000  177604  ERR01:  MOV     #0,ERCOUNT      ;SET UP ERROR COUNT 0
        000001                          N=N+1
001260  004567  007252                  JSR     X5,STAER        ;REPORT STATIC ERROR
001264  105767  177604          XSTA11: TSTB    WRD             ;TEST FOR READY
001270  001005                          BNE     LPST1           ;BRANCH IF READY
001272  012767  000001  177564  ERR11:  MOV     #1,ERCOUNT      ;SET UP ERROR COUNT 1
        000002                          N=N+1
001300  004567  007232                  JSR     X5,STAER        ;REPORT READY NOT SET
001304  000004                  LPST1:  SCOPE                   ;ENTER SCOPE LOOP
001306  001232                          STA11
                                        ;TEST CURRENT ADDRESS REGISTER
                                        ;
001310  000005                  STA12:  RESET                   ;CLEAR THE WORLD
001312  017767  177502  177554          MOV     @RKBA,WRD       ;FETCH RKBA REGISTER
001320  001405                          BEQ     LPST2           ;REGISTER OK
001322  012767  000002  177534  ERR21:  MOV     #2,ERCOUNT      ;SET UP ERROR COUNT 2
        000003                          N=N+1
001330  004567  007202                  JSR     X5,STAER        ;REPORT ERROR
001334  000004                  LPST2:  SCOPE                   ;ENTER SCOPE LOOP
```

```
                                        ST.A12
                                ;
                                ;TEST WORD COUNT REGISTER
                                ;
001340  000005                STA13:  RESET                   ;CLEAR THE WORLD
001342  017767  177450  177524        MOV     @RKWC,WRD       ;FETCH WC; REG;
001350  001405                        BEQ     LPST3           ;REGISTER OK
001352  012767  000003  177504  ERR3: MOV     #3,ERCOUNT      ;SET UP ERROR COUNT 3
        000004                        N=N+1
001360  004567  007152                JSR     X5,STAER        ;REPORT ERROR
001364  000004                LPST3:  SCOPE                   ;ENTER SCOPE LOOP
001366  001340                        STA13
```

```
                                        ;
                                        ;TEST DISK ADDRESS REGISTER
                                        ;
001370  000005          STAI4:  RESET                   ;CLEAR THE WORLD
001372  017767  177424  177474          MOV     @RKDA,WRD       ;FETCH RKDA REGISTER
001400  001405                  BEQ     LPST4           ;REGISTER CLEARED
001402  012767  000004  177454 ERR4:   MOV     #4,ERCOUNT      ;SET UP ERROR COUNT 4
        000005                  N=N+1
001410  004567  007122                  JSR     %5,STAER        ;REPORT ERROR
001414  000004          LPST4:  SCOPE                   ;ENTER SCOPE LOOP
001416  001370                  STAI4
                                        ;
                                        ;TEST CONTROL ERROR REGISTER
                                        ;
001420  000005          STAI5:  RESET                   ;CLEAR THE WORLD
001422  017767  177376  177444          MOV     @RKER,WRD       ;FETCH RKER REGISTERS
001430  001405                  BEQ     LPST5           ;RKER CLEARED
001432  012767  000005  177424 ERR5:   MOV     #5,ERCOUNT      ;SET UP ERROR COUNT 5
        000006                  N=N+1
001440  004567  007072                  JSR     %5,STAER        ;REPORT ERROR
001444  000004          LPST5:  SCOPE                   ;ENTER SCOPE LOOP
001446  001420                  STAI5
                                        ;
                                        ;TEST DATA BUFFER REGISTER
                                        ;
001450  000005          STAI6:  RESET                   ;CLEAR THE WORLD
001452  017767  177352  177414          MOV     @RKDB,WRD       ;FETCH RKDB REGISTER
001460  001405                  BEQ     LPST6           ;RKDB CLEARED
001462  012767  000006  177374 ERR6:   MOV     #6,ERCOUNT      ;SET UP ERROR COUNT 6
        000007                  N=N+1
001470  004567  007042                  JSR     %5,STAER        ;REPORT ERROR
001474  000004          LPST6:  SCOPE                   ;ENTER SCOPE LOOP
001476  001450                  STAI6
                                        ;
                                        ;TEST RKMRINTENANCE REGISTER
                                        ;
001500  000005          STAI7:  RESET                   ;CLEAR THE WORLD
001502  017767  177324  177364          MOV     @RKMR,WRD       ;FETCH RKMRINTENANCE REG
001510  001405                  BEQ     LPST7           ;RKMR CLEARED
001512  012767  000007  177344 ERR7:   MOV     #7,ERCOUNT      ;SET UP ERROR COUNT 7
        000010                  N=N+1
001520  004567  007012                  JSR     %5,STAER        ;REPORT ERROR
001524  000004          LPST7:  SCOPE                   ;ENTER SCOPE LOOP
001526  001500                  STAI7
                                        ;
                                        ;CAN WE SET THE FUNCTION BITS IN THE RKCS REG,
                                        ;BITS 7,6,5,4,3,2&1
                                        ;
001530  012777  000340  177244 STAI10:  MOV     #340,@PS
001536  112777  000176  177250          MOVB    #176,@RKCS      ;SET DISK FUNCTION BITS
001544  017767  177244  177322          MOV     @RKCS,WRD       ;FETCH FUNCTION BITS
001552  022767  000376  177314          CMP     #376,WRD        ;ARE THE FUNCTION BITS SET
001560  001410                  BEQ     LPST10          ;FUNCTION BITS SET
```

```
  282  001562  012767  000376  177306          MOV     #376,WRD1       ;SET UP FOR ERROR REPORT
  283  001570  012767  000010  177266  ERR10:  MOV     #10,ERCOUNT     ;SET UP ERROR COUNT 10
  284          000011                          N=N+1
  285  001576  004567  007006                  JSR     %5,STAER1       ;REPORT ERROR (ERROR IN FUNCTION BITS)
  286  001602  000004                  LPST10: $SCOPE                  ;ENTER SCOPE LOOP
  287  001604  001530                          STAI10
```

```
                                        ;
                                        ;WILL DISK CLEAR CLEAR THE FUNCTION BITS
                                        ;
001606  012777  000340  177166  STAI11: MOV     #340,@PS
001614  012777  000176  177172          MOV     #176,@RKCS      ;SET DISK FUNCTION BITS
001622  012777  000001  177164          MOV     #BIT0,@RKCS     ;EXECUTE DISK CLEAR
001630  105777  177160                  TSTB    @RKCS
001634  100375                          BPL     .-4
001636  017767  177152  177230          MOV     @RKCS,WRD       ;FETCH CONTROL REG
001644  022767  000200  177222          CMP     #200,WRD        ;IS ONLY READY SET
001652  001405                          BEQ     LPST11          ;REGISTER CLEARED
001654  012767  000011  177202  ERR11:  MOV     #11,ERCOUNT     ;SET UP ERROR COUNT 11
        000012                          N=N+1
001662  004567  006650                  JSR     %5,STAER        ;REPORT ERROR
001666  000004                  LPST11: SCOPE                   ;ENTER SCOPE LOOP
001670  001606                          STAI11
                                        ;
                                        ;CAN WE SET ALL THE RKBA BITS
                                        ;
001672  012767  177777  177176  STAI12: MOV     #177777,WRD1    ;SET UP CURRENT ADDR; OF ALL ONES
001700  016777  177172  177112          MOV     WRD1,@RKBA      ;LOAD RKBA
001706  017767  177106  177160          MOV     @RKBA,WRD       ;FETCH RKBA
001714  026767  177156  177152          CMP     WRD1,WRD        ;COMPARE FOR ALL BITS SET
001722  001405                          BEQ     LPST12          ;ALL BITS SET?
001724  012767  000012  177132  ERR12:  MOV     #12,ERCOUNT     ;SET UP ERROR COUNT 12
        000013                          N=N+1
001732  004567  006652                  JSR     %5,STAER1       ;REPORT ERROR
001736  000004                  LPST12: SCOPE                   ;ENTER SCOPE LOOP
001740  001672                          STAI12
                                        ;
                                        ;WILL DISK CLEAR * CLEAR THE RKBA REGISTER
                                        ;
001742  012777  177777  177050  STAI13: MOV     #177777,@RKBA   ;SET RKBA EQUAL TO ALL ONES
001750  012777  000001  177036          MOV     #BIT0,@RKCS     ;EXECUTE DISK CLEAR
001756  105777  177032                  TSTB    @RKCS
001762  100375                          BPL     .-4
001764  005777  177030                  TST     @RKBA           ;TEST FOR BIT0 SET IN RKBA (READ ONLY BIT)
001770  001410                          BEQ     LPST13          ;WAS THE REST OF RKBA CLEARED?
001772  017767  177016  177074          MOV     @RKCS,WRD       ;NO! FETCH RKBA REG,
002000  012767  000013  177056  ERR13:  MOV     #13,ERCOUNT     ;SET UP ERROR COUNT 13
        000014                          N=N+1
002006  004567  006524                  JSR     %5,STAER        ;REPORT ERROR
002012  000004                  LPST13: SCOPE                   ;ENTER SCOPE LOOP
002014  001742                          STAI13
                                        ;
                                        ;
                                        ;CAN WE SET ALL WORD COUNT BITS
                                        ;
002016  012767  177777  177052  STAI14: MOV     #177777,WRD1
002024  016777  177046  176764          MOV     WRD1,@RKWC      ;SET WORD COUNT EQUAL TO ALL ONES
002032  017767  176760  177034          MOV     @RKWC,WRD       ;FETCH WORD COUNT
002040  026767  177032  177026          CMP     WRD1,WRD        ;IS WORD COUNT EQUAL TO ALL ONES
002046  001405                          BEQ     LPST14          ;WORD COUNT EQUAL TO ALL ONES
```

```
   342  002050  012767  000014  177006  ERR14:  MOV     #14,ERRCOUNT    ;SET UP ERROR COUNT 14
   343          000015                  N=N+1
   344  002056  004567  006526                  JSR     X5,STAER1       ;REPORT WORD COUNT NOT EQUAL TO ALL ONES
   345  002062  000004                  LPST14: SCOPE
   346  002064  002016                          STA114
```

```
                                        ;
                                        ;WILL DISK CLEAR-CLEAR THE WORD COUNT REGISTER
                                        ;
002066  012777  177777  176722  STAI15: MOV     #177777,@RKWC   ;SET WC: REGISTER EQUAL TO ALL ONES
002074  012777  000001  176712          MOV     #BIT0,@RKCS     ;EXECUTE DISK CLEAR
002102  105777  176706                  TSTB    @RKCS
002106  100375                          BPL     .-4
002110  017767  176702  176756          MOV     @RKWC,WRD       ;FETCH WORD COUNT REGISTER
002116  001405                          BEQ     LPST15          ;YES: EXIT
002120  012767  000015  176736  ERR15:  MOV     #15,ERCOUNT     ;SET UP ERROR COUNT 15
        000016                          N=N+1
002126  004567  006404                  JSR     R5,STAER        ;REPORT ERROR
002132  000004          LPST15: SCOPE                           ;ENTER SCOPE LOOP
002134  002066                          STAI15
                                        ;
                                        ;CAN WE SET ALL THE BITS IN THE RKDA REGISTER:
                                        ;
002136  012767  177777  176732  STAI16: MOV     #177777,WRD1
002144  016777  176726  176650          MOV     WRD1,@RKDA      ;SET RKDA TO ALL ONES
002152  017767  176644  176714          MOV     @RKDA,WRD       ;FETCH RKDA REGISTER
002160  026767  176712  176706          CMP     WRD1,WRD        ;ARE ALL BITS SET
002166  001405                          BEQ     LPST16          ;YES: EXIT
002170  012767  000016  176666  ERR16:  MOV     #16,ERCOUNT     ;SET UP ERROR COUNT 16
        000017                          N=N+1
002176  004567  006406                  JSR     R5,STAER1       ;NOT ALL BITS SET REPORT ERROR
002202  000004          LPST16: SCOPE                           ;ENTER SCOPE LOOP
002204  002136                          STAI16
                                        ;
                                        ;CAN WE CLEAR THE RKDA REG' WITH DISK CLEAR.
                                        ;
002206  012777  177777  176606  STAI17: MOV     #177777,@RKDA   ;SET RKDA TO ALL ONES
002214  012777  000001  176572          MOV     #BIT0,@RKCS     ;EXECUTE DISK CLEAR
002222  105777  176566                  TSTB    @RKCS
002226  100375                          BPL     .-4
002230  005777  176566                  TST     @RKDA           ;TEST FOR ZERO RKDA
002234  001410                          BEQ     LPST17          ;YES EXIT
002236  017767  176560  176630          MOV     @RKDA,WRD       ;NO BITS SET IN RKDA
002244  012767  000017  176612  ERR17:  MOV     #17,ERCOUNT     ;SET UP ERROR COUNT 17
        000020                          N=N+1
002252  004567  006260                  JSR     R5,STAER        ;REPORT ERROR
002256  000004          LPST17: SCOPE                           ;ENTER SCOPE LOOP
002260  002206                          STAI17
                                        ;
```

391

```
                                        ;
                                        ;DO ONE WORD WRITE FOLLOWED BY ONE WORD WRITE CONTINUE
                                        ;ERROR1=BUSY FAILED TO SET
                                        ;ERROR2=BUSY FAILED TO CLEAR
                                        ;ERROR3=CONTROL ERROR WHEN X-FERING DATA
                                        ;ERROR5=DAR FAILED TO INCREMENT WHEN X-FERING DATA
                                        ;ERROR6=WORD COUNT FAILED TO OVERFLOW
                                        ;ERROR7=CMA FAILED TO INCREMENT
                                        ;
                                        ;
                                        ;* * * EXECUTE THE ONE WORD WRITE* * *
002262  012777  000001  176524  STAI22: MOV     #BIT0,@RKCS     ;CLEAR THE DISK WORLD
002270  105777  176520                  TSTB    @RKCS
002274  100375                         BPL     .-4
002276  012767  177777  007252          MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
002304  012777  011556  176506          MOV     #OUTBUF,@RKBA   ;SET UP CURRENT ADDRESS
002312  012777  177777  176476          MOV     #-1,@RKWC       ;SET WORD COUNT TO -1
002320  052777  000003  176466          BIS     #3,@RKCS        ;GO WRITE
002326  105777  176462                  TSTB    @RKCS           ;TEST FOR RDY=0
002332  100011                         BPL     STAI23          ;RDY=0
002334  017767  176454  176532          MOV     @RKCS,WRD       ;BUSY NOT SET; FETCH RKCS
002342                                  ERROR1;
002342  012767  000020  176514  ERR20:  MOV     #20,ERCOUNT     ;SET UP ERROR COUNT 20
        000021                          N=N+1
002350  004567  006162                  JSR     X5,STAER        ;REPORT ERROR
002354  000742                         BR      STAI22          ;RESTART TEST
002356  005067  176512          STAI23: CLR     WRD
002362  005267  176506          INCWAT: INC     WRD             ;WAIT FOR BUSY=0
002366  105777  176422                  TSTB    @RKCS           ;IS BUSY CLEARED
002372  100417                         BMI     STAI24          ;FLAG CLEARED
002374  005767  176474                  TST     WRD             ;HAVE WE WAITED LONG ENOUGH
002400  001370                         BNE     INCWAT          ;RDY FAILED TO SET
002402  017767  176406  176464          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS REG
002410  052767  000002  176460          BIS     #2,WRD1 ;WANT RKCS SHOULD CONTAIN
002416                                  ERROR2;
002416  012767  000021  176440  ERR21:  MOV     #21,ERCOUNT     ;SET UP ERROR COUNT 21
        000022                          N=N+1
002424  004567  006160                  JSR     X5,STAER1       ;REPORT ERROR
002430  000714                         BR      STAI22          ;RESTART TEST
002432  017767  176356  176434  STAI24: MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS REG
002440  005767  176430                  TST     WRD             ;IS ERROR FLAG SET
002444  100012                         BPL     STAI26          ;NO; X-FER OK
002446  012767  000202  176422          MOV     #202,WRD1       ;WHAT RKCS SHOULD CONTAIN
002454                                  ERROR3;
002454  012767  000022  176402  ERR22:  MOV     #22,ERCOUNT     ;SET UP ERROR COUNT 22
        000023                          N=N+1
002462  004567  006122                  JSR     X5,STAER1       ;REPORT ERROR
002466  000167  177570                  JMP     STAI22          ;RESTART TEST
002472  017767  176324  176374  STAI26: MOV     @RKDA,WRD       ;WAS RKDA INCREMENTED BY 1
002500  022767  000001  176366          CMP     #BIT0,WRD       ;IS RKDA CORRECT
002506  001412                         BEQ     STAI27          ;RKDA OK
002510  012767  000001  176360          MOV     #BIT0,WRD1      ;WHAT RKDA SHOULD CONTAIN
002516                                  ERROR5;
```

```
446  002516  012767  000023  170340  ERR23:  MOV     #23,ENCOUNT     ;SET UP ERROR COUNT 23
447          000024                  N=N+1
448  002524  004567  006060                  JSR     X5,STAER1       ;REPORT ERROR
449  002530  000167  177526                  JMP     STA122          ;RESTART TEST
```

```
                                                ;
002534  017767  176256  176332  STA127: MOV     @RKWC,WRD       ;FETCH WORD COUNT
002542  001407                          BEQ     STA130          ;WORD COUNT DID OVERFLOW
002544                                  ERR06:
002544  012767  000024  176312  ERR24:  MOV     #24,ERCOUNT     ;SET UP ERROR COUNT 24
                000025                  N=N+1
002552  004567  005760                  JSR     %5,STAER        ;WORD COUNT FAILED TO OVERFLOW
002556  000167  177500                  JMP     STA122          ;RESTART TEST
002562  017767  176232  176304  STA130: MOV     @RKBA,WRD       ;FETCH CURRENT ADDRESS
002570  012767  011500  176300          MOV     #OUTBUF+2,WRD1  ;WHAT RKBA SHOULD EQUAL
002576  026767  176274  176270          CMP     WRD1,WRD        ;IS RKBA CORRECT
002604  001407                          BEQ     STA131          ;YES EXECUTE CONTINUE
002606                                  ERR07:
002606  012767  000025  176250  ERR25:  MOV     #25,ERCOUNT     ;SET UP ERROR COUNT 25
                000026                  N=N+1
002614  004567  005770                  JSR     %5,STAER1       ;REPORT ERROR IN RKBA
002620  000167  177436                  JMP     STA122          ;ERROR RESTART TEST
```

```
                                        ;       TEST WRITE
                                        ;A WRITE CONTINUE WILL BE EXECUTED NOW
                                        ;WORD COUNT WILL BE SET TO -1 AGAIN
                                        ;
                                        ;ERROR10=BUSY NOT SET BY GO
                                        ;ERROR11=BUSY NOT CLEARED BY OVERFLOW
                                        ;ERROR12=DISK ERROR OCCURED WHILE X-FERING
                                        ;ERROR14=DAR DID NOT EQUAL 2 AFTER SECOND X-FER
                                        ;ERROR15=WC DID NOT OVERFLOW AT THE END OF X-FER
                                        ;ERROR16=CMA DID NOT EQUAL OUTBUF+2 AT END OF X-FER
                                        ;
                                        ;
002624  012777  177777  176164  STAI31: MOV     #177777,@RKWC   ;SET WC: TO -1
002632  052777  000001  176154          BIS     #BIT0,@RKCS     ;SET GO TO CONTINUE
002640  105777  176150                  TSTB    @RKCS           ;TEST FOR RDY SET
002644  100012                          BPL     STAI32          ;BUSY WAS SET BY GO
002646  017767  176142  176220          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
002654                          ERROR10:
002654  012767  000026  176202  ERR26: MOV     #26,ERCOUNT     ;SET UP ERROR COUNT 26
        000027                          N=N+1
002662  004567  005650                  JSR     %5,STAER        ;REPORT BUSY NOT SET
002666  000167  177370                  JMP     STAI22
002672  005067  176176          STAI32: CLR     WRD
002676  105777  176112          INCBUSY:TSTB    @RKCS           ;TST FOR RDY SET BY OVERFLO
002702  100415                          BMI     STAI33          ;READY SET CONTINUE
002704  005267  176164                  INC     WRD             ;WAIT FOR RDY=1
002710  001372                          BNE     INCBUSY         ;GO WAIT FOR RDY
002712  017767  176076  176154          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
002720                          ERROR11:
002720  012767  000027  176136  ERR27: MOV     #27,ERCOUNT     ;SET UP ERROR COUNT 27
        000030                          N=N+1
002726  004567  005604                  JSR     %5,STAER        ;REPORT BUSY NOT CLEARED
002732  000167  177324                  JMP     STAI22          ;RESTART ROUTINE
002736  005777  176052          STAI33: TST     @RKCS           ;DID AN ERROR OCCUR WHILE X-FERING
002742  100015                          BPL     STAI35          ;NO CONTINUE
002744  017767  176044  176122          MOV     @RKCS,WRD       ;YES! CONTENTS OF RKCS
002752  017767  176046  176116          MOV     @RKER,WRD1      ;EXT ERROR BITS
002760                          ERROR12:
002760  012767  000030  176076  ERR30: MOV     #30,ERCOUNT     ;SET UP ERROR COUNT 30
        000031                          N=N+1
002766  004567  005616                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED;
002772  000167  177264                  JMP     STAI22          ;RESTART ROUTINE
002776  017767  176020  176070  STAI35: MOV     @RKDA,WRD       ;DID RKDA INCREMENT ON CONTINUE
003004  012767  000002  176064          MOV     #2,WRD1         ;WHAT RKDA SHOULD CONTAIN
003012  026767  176060  176054          CMP     WRD1,WRD        ;IS RKDA CORRECT
003020  001407                          BEQ     STAI36          ;RKDA OK
003022                          ERROR14:
003022  012767  000031  176034  ERR31: MOV     #31,ERCOUNT     ;SET UP ERROR COUNT 31
        000032                          N=N+1
003030  004567  005554                  JSR     %5,STAER1       ;REPORT RKDA INCORRECT
003034  000167  177222                  JMP     STAI22          ;RE-START ROUTINE
003040  017767  175752  176026  STAI36: MOV     @RKWC,WRD       ;FETCH WORD COUNT
003046  001407                          BEQ     STAI37          ;WORD COUNT OVERFLOWED
```

```
                                        ERROR15:
003050                                  ERR32:  MOV     #32,ERCOUNT     ;SET UP ERROR COUNT 32
                000033                          N=N+1
003056  004567  005454                          JSR     %5,STAER        ;REPORT WORD COUNT FAILED TO CLEAR
003062  000167  177174                          JMP     STA122          ;RESTART ROUTINE
003066  017767  175726  176000        STA137:   MOV     @RKBA,WRD       ;FETCH RKBA
003074  012767  011562  175774                  MOV     #OUTBUF+4,WRD1  ;WHAT RKBA SHOULD EQUAL
003102  026767  175770  175764                  CMP     WRD1,WRD        ;IS RKBA CORRECT
003110  001407                                  BEQ     LPST22          ;;RKBA WAS CORRECT
003112                                  ERROR16:
003112  012767  000033  175744          ERR33:  MOV     #33,ERCOUNT     ;SET UP ERROR COUNT 33
                000034                          N=N+1
003120  004567  005404                          JSR     %5,STAER1       ;REPORT RKBA INCORRECT
003124  000167  177132                          JMP     STA122          ;RESTART ROUTINE
003130  000004                        LPST22:   SCOPE                   ;ENTER SCOPE LOOP
003132  002262                                  STA122
```

```
                                        ;
                                        ;              TEST WRITE CHECK
                                        ;DO A ONE WORD WRITE CHECK FOLLOWED BY
                                        ;A ONE WORD WRITE CHECK CONTINUE
                                        ;
                                        ;ERROR17=BUSY FAILED TO SET
                                        ;ERROR20=BUSY FAILED TO CLEAR
                                        ;ERROR21=DISK ERROR WHEN X-FERING DATA
                                        ;ERROR23=DAR FAILED TO INCREMENT WHEN X-FERING DATA
                                        ;ERROR24=WORD COUNT FAILED TO OVERFLOW
                                        ;ERROR25=CMA FAILED TO INCREMENT
                                        ;
                                        ;
                                        ;* * *EXECUTE THE ONE WORD WRITE CHECK* * *
003134  012777  000001  175652  STAI40: MOV     #BIT0,@RKCS     ;CLEAR THE DISK WORLD
003142  105777  175646                  TSTB    @RKCS
003146  100375                          BPL     .-4
003150  012767  177777  000400          MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
003156  012777  011556  175634          MOV     #OUTBUF,@RKBA   ;SET UP CURRENT ADDRESS
003164  012777  177777  175624          MOV     #-1,@RKWC       ;SET WORD COUNT TO -1
003172  052777  000007  175614          BIS     #7,@RKCS        ;GO WRITE CHECK
003200  105777  175610                  TSTB    @RKCS           ;TEST FOR READY
003204  100011                          BPL     STAI41          ;NOT READY
003206  017767  175602  175660          MOV     @RKCS,WRD       ;SET: FETCH RKCS
003214                          ERROR17:
003214  012767  000034  175642  ERR34:  MOV     #34,ERCOUNT     ;SET UP ERROR COUNT 34
        000035                          N=N+1
003222  004567  005310                  JSR     X5,STAER        ;REPORT ERROR
003226  000742                          BR      STAI40          ;RESTART TEST
003230  005067  175640          STAI41: CLR     WRD
003234  005267  175634          WCWAT:  INC     WRD             ;WAIT FOR READY
003240  105777  175550                  TSTB    @RKCS
003244  100417                          BMI     STAI42          ;FLAG SET
003246  005767  175622                  TST     WRD             ;HAVE WE WAITED LONG ENOUGH
003252  001370                          BNE     WCWAT           ;READY FAILED TO SET
003254  017767  175534  175612          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS REG
003262  052767  000000  175606          BIS     #0,WRD1 ;WANT RKCS SHOULD CONTAIN
003270                          ERROR20:
003270  012767  000035  175566  ERR35:  MOV     #35,ERCOUNT     ;SET UP ERROR COUNT 35
        000036                          N=N+1
003276  004567  005306                  JSR     X5,STAER1       ;REPORT ERROR
003302  000714                          BR      STAI40          ;RESTART TEST
003304  017767  175504  175562  STAI42: MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS REG
003312  100012                          BPL     STAI44          ;NO: X-FER OK
003314  012767  000200  175554          MOV     #200,WRD1
003322                          ERROR21:
003322  012767  000036  175534  ERR36:  MOV     #36,ERCOUNT     ;SET UP ERROR COUNT 36
        000037                          N=N+1
003330  004567  005254                  JSR     X5,STAER1       ;REPORT ERROR
003334  000167  177774                  JMP     STAI40          ;RESTART TEST
003340  017767  175456  175526  STAI44: MOV     @RKDA,WRD       ;WAS RKDA INCREMENTED BY 1
003346  022767  000001  175520          CMP     #BIT0,WRD       ;IS RKDA CORRECT
003354  001412                          BEQ     STAI45          ;RKDA OK
003356  012767  000001  175512          MOV     #BIT0,WRD1      ;WHAT RKDA SHOULD CONTAIN
```

```
  591  003364                          ERROR23:
  592  003364  012767  000037  175472  ERR37:  MOV     #37,ERCOUNT     ;SET UP ERROR COUNT 37
  593          000040                          N=N+1
  594  003372  004567  005212                  JSR     %5,STAER1       ;REPORT ERROR
  595  003376  000167  177532                  JMP     STA140          ;RESTART TEST
  596  003402  017767  175410  175464  STA145: MOV     @RKWC,WRD       ;FETCH WORD COUNT
  597  003410  001407                          BEQ     STA146          ;WORD COUNT DID OVERFLOW
  598  003412                          ERROR24:
  599  003412  012767  000040  175444  ERR40:  MOV     #40,ERCOUNT     ;SET UP ERROR COUNT 40
  600          000041                          N=N+1
  601  003420  004567  005112                  JSR     %5,STAER        ;WORD COUNT FAILED TO OVERFLOW
  602  003424  000167  177504                  JMP     STA140          ;RESTART TEST
  603  003430  017767  175364  175436  STA146: MOV     @RKBA,WRD       ;FETCH CURRENT ADDRESS
  604  003436  012767  011560  175432          MOV     #OUTBUF+2,WRD1  ;WHAT RKBA SHOULD EQUAL
  605  003444  026767  175426  175422          CMP     WRD1,WRD        ;IS RKBA CORRECT
  606  003452  001407                          BEQ     STA147          ;YES EXECUTE CONTINUE
  607  003454                          ERROR25:
  608  003454  012767  000041  175402  ERR41:  MOV     #41,ERCOUNT     ;SET UP ERROR COUNT 41
  609          000042                          N=N+1
  610  003462  004567  005122                  JSR     %5,STAER1       ;REPORT ERROR IN RKBA
  611  003466  000167  177442                  JMP     STA140          ;ERROR RESTART TEST
```

```
                                        ;
                                        ;A WRITE CHECK CONTINUE CONTINUE WILL BE EXECUTED NOW
                                        ;WORD COUNT WILL BE SET TO -1 AGAIN
                                        ;
                                        ;ERROR26=BUSY WAS NOT SET BY GO
                                        ;ERROR27=BUSY NOT CLEARED BY OVERFLOW
                                        ;ERROR30=DISK ERROR OCCURRED WHILE X-FERING
                                        ;ERROR32=DAR DID NOT EQUAL 2 AFTER SECOND X-FER
                                        ;ERROR33=WC DID NOT=0 AFTER X-FER
                                        ;ERROR34=CMA DID NOT=OUTBUF+2 AT END OF X-FER
                                        ;
003472  012777  177777  175316  STAI47: MOV     #177777,@RKWC   ;SET WC: TO -1
003500  052777  000001  175306          BIS     #BIT0,@RKCS     ;SET GO TO CONTINUE
003506  105777  175302                  TSTB    @RKCS           ;TEST FOR READY
003512  100012                          BPL     STAI50          ;
003514  017767  175274  175352          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
003522                          ERROR26:
003522  012767  000042  175334  ERR42:  MOV     #42,ERCOUNT     ;SET UP ERROR COUNT 42
        000043                  N=N+1
003530  004567  005002                  JSR     %5,STAER        ;REPORT BUSY NOT SET
003534  000167  177374                  JMP     STAI40
003540  005067  175330          STAI50: CLR     WRD
003544  105777  175244          WCBUSI: TSTB    @RKCS           ;TST FOR RDY SET BY OVERFLOW
003550  100415                          BMI     STAI51          ;BRANCH IF READY SET
003552  005267  175316                  INC     WRD             ;WAIT FOR BUSY=0
003556  001372                          BNE     WCBUSI          ;GO WAIT FOR RDY
003560  017767  175230  175306          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
003566                          ERROR27:
003566  012767  000043  175270  ERR43:  MOV     #43,ERCOUNT     ;SET UP ERROR COUNT 43
        000044                  N=N+1
003574  004567  004736                  JSR     %5,STAER        ;REPORT RDY NOT SET
003600  000167  177330                  JMP     STAI40          ;RESTART ROUTINE
003604  005777  175204          STAI51: TST     @RKCS           ;DID AN ERROR OCCUR WHILE X-FERING
003610  100015                          BPL     STAI53          ;NO CONTINUE
003612  017767  175176  175254          MOV     @RKCS,WRD       ;YES! CONTENTS OF RKCS
003620  012767  000206  175250          MOV     #206,WRD1
003626                          ERROR30:
003626  012767  000044  175230  ERR44:  MOV     #44,ERCOUNT     ;SET UP ERROR COUNT 44
        000045                  N=N+1
003634  004567  004750                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED
003640  000167  177270                  JMP     STAI40          ;RESTART ROUTINE
003644  017767  175152  175222  STAI53: MOV     @RKDA,WRD       ;DID RKDA INCREMENT ON CONTINUE
003652  012767  000002  175216          MOV     #2,WRD1         ;WHAT RKDA SHOULD CONTAIN
003660  026767  175212  175206          CMP     WRD1,WRD        ;IS RKDA CORRECT
003666  001407                          BEQ     STAI54          ;RKDA OK
003670                          ERROR32:
003670  012767  000045  175166  ERR45:  MOV     #45,ERCOUNT     ;SET UP ERROR COUNT 45
        000046                  N=N+1
003676  004567  004706                  JSR     %5,STAER1       ;REPORT RKDA INCORRECT
003702  000167  177226                  JMP     STAI40          ;RE-START ROUTINE
003706  017767  175104  175160  STAI54: MOV     @RKWC,WRD       ;FETCH WORD COUNT
003714  001407                          BEQ     STAI55          ;WORD COUNT OVERFLOWED
```

```
666  003716                                ERROR33:
667  003716  012767  000046  175140  ERR46:  MOV     #46,ERCOUNT     ;SET UP ERROR COUNT 46
668          000047                        N=N+1
669  003724  004567  004606                 JSR     %5,STAER        ;REPORT WORD COUNT FAILED TO CLEAR
670  003730  000167  177200                 JMP     STAI40          ;RESTART ROUTINE
671  003734  017767  175060  175132  STAI55: MOV     @RKBA,WRD       ;FETCH RKBA
672  003742  012767  011562  175126        MOV     #OUTBUF+4,WRD1  ;WHAT RKBA CHOULD EQUAL
673  003750  026767  175122  175116        CMP     WRD1,WRD        ;IS RKBA CORRECT
674  003756  001407                        BEQ     LPST40          ;;RKBA WAS CORRECT
675  003760                                ERROR34:
676  003760  012767  000047  175076  ERR47:  MOV     #47,ERCOUNT     ;SET UP ERROR COUNT 47
677          000050                        N=N+1
678  003766  004567  004616                 JSR     %5,STAER1       ;REPORT RKBA INCORRECT
679  003772  000167  177136                 JMP     STAI40          ;RESTART ROUTINE
680  003776  000004                LPST40: SCOPE                   ;ENTER SCOPE LOOP
681  004000  003134                        STAI40
```

```
                                        ;       TEST READ
                                        ;ONE WORD READ FOLLOWED BY A ONE WORD
                                        ;READ CONTINUE,
                                        ;
                                        ;ERROR35=BUSY WAS NOT SET BY GO
                                        ;ERROR36=BUSY NOT CLEARED BY OVERFLOW
                                        ;ERROR37=DISK ERROR OCCURRED WHILE X-FERING DATA
                                        ;ERROR41=DAR FAILED TO INCREMENT WHEN X-FERING DATA
                                        ;ERROR42=WORD COUNT FAILED TO OVERFLOW
                                        ;ERROR43=CMA FAILED TO INCREMENT
                                        ;
                                        ;
004002  012777  000001  175004  STA156: MOV     #BIT0,@RKCS     ;CLEAR THE DISK WORLD
004010  105777  175000                  TSTB    @RKCS
004014  100375                          BPL     .-4
004016  012767  177777  005532          MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
004024  012777  011556  174766          MOV     #OUTBUF,@RKBA   ;SET UP CURRENT ADDRESS
004032  012777  177777  174756          MOV     #-1,@RKWC               ;SET WORD COUNT TO -1
004040  052777  000005  174746          BIS     #5,@RKCS                ;GO READ
004046  105777  174742                  TSTB    @RKCS           ;TEST FOR BUSY=1
004052  100011                          BPL     STA157          ;BUSY SET
004054  017767  174734  175012          MOV     @RKCS,WRD       ;BUSY NOT SET; FETCH RKCS
004062                          ERROR35:
004062  012767  000050  174774  ERR501: MOV     #50,ERCOUNT     ;SET UP ERROR COUNT 50
                000051                  N=N+1
004070  004567  004442                  JSR     %5,STAER        ;REPORT ERROR
004074  000742                          BR      STA150          ;RESTART TEST
004076  005067  174772          STA157: CLR     WRD
004102  005267  174766          RDWAT1: INC     WRD             ;WAIT FOR BUSY=0
004106  105777  174702                  TSTB    @RKCS           ;IS BUSY CLEARED
004112  100417                          BMI     STA160          ;FLAG CLEARED
004114  005767  174754                  TST     WRD             ;HAVE WE WAITED LONG ENOUGH
004120  001370                          BNE     RDWAT           ;BUSY FAILED TO CLEAR
004122  017767  174666  174744          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS REG
004130  052767  000005  174740          BIS     #5,WRQ1 ;WANT RKCS SHOULD CONTAIN
004136                          ERROR36:
004136  012767  000051  174720  ERR511: MOV     #51,ERCOUNT     ;SET UP ERROR COUNT 51
                000052                  N=N+1
004144  004567  004440                  JSR     %5,STAER1       ;REPORT ERROR
004150  000714                          BR      STA150          ;RESTART TEST
004152  017767  174636  174714  STA160: MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS REG
004160  005767  174710                  TST     WRD             ;IS ERROR FLAG SET
004164  100012                          BPL     STA162          ;NO! X-FER OK
004166  012767  000003  174702          MOV     #3,WRQ1 ;WHAT RKCS SHOULD CONTAIN
004174                          ERROR37:
004174  012767  000052  174662  ERR521: MOV     #52,ERCOUNT     ;SET UP ERROR COUNT 52
                000053                  N=N+1
004202  004567  004402                  JSR     %5,STAER1       ;REPORT ERROR
004206  000167  177570                  JMP     STA150          ;RESTART TEST
004212  017767  174604  174654  STA162: MOV     @RKDA,WRD       ;WAS RKDA INCREMENTED BY 1
004220  022767  000001  174646          CMP     #BIT0,WRD       ;IS RKDA CORRECT
004226  001412                          BEQ     STA163          ;RKDA OK
```

```
  736  004230  012767  000001  174640          MOV     #BIT0,WRD1      ;WHAT RKDA SHOULD CONTAIN
  737  004236                          ERROR41:
  738  004236  012767  000053  174620  ERR53:  MOV     #53,ERCOUNT     ;SET UP ERROR COUNT 53
  739          000054                  N=N+1
  740  004244  004567  004340                  JSR     %5,STAER1       ;REPORT ERROR
  741  004250  000167  177526                  JMP     STAI50          ;RESTART TEST
  742  004254  017767  174536  174612  STAI63: MOV     @RKWC,WRD       ;FETCH WORD COUNT
  743  004262  001407                          BEQ     STAI64          ;WORD COUNT DID OVERFLOW
  744  004264                          ERROR42:
  745  004264  012767  000054  174572  ERR54:  MOV     #54,ERCOUNT     ;SET UP ERROR COUNT 54
  746          000055                  N=N+1
  747  004272  004567  004240                  JSR     %5,STAER        ;WORD COUNT FAILED TO OVERFLOW
  748  004276  000167  177500                  JMP     STAI50          ;RESTART TEST
  749  004302  017767  174512  174564  STAI64: MOV     @RKBA,WRD       ;FETCH CURRENT ADDRESS
  750  004310  012767  011560  174560          MOV     #OUTBUF+2,WRD1  ;WHAT RKBA SHOULD EQUAL
  751  004316  026767  174554  174550          CMP     WRD1,WRD        ;IS RKBA CORRECT
  752  004324  001407                          BEQ     $TAI62          ;YES EXECUTE CONTINUE
  753  004326                          ERROR43:
  754  004326  012767  000055  174530  ERR55:  MOV     #55,ERCOUNT     ;SET UP ERROR COUNT 55
  755          000056                  N=N+1
  756  004334  004567  004250                  JSR     %5,STAER1       ;REPORT ERROR IN RKBA
  757  004340  000167  177436                  JMP     STAI50          ;ERROR RESTART TEST
```

```
                                        ;A READ CONTINUE WILL BE EXECUTED NOW
                                        ;WORD COUNT WILL BE SET TO -1 AGAIN
                                        ;
                                        ;EROR44=BUSY NOT SET BY GO
                                        ;EBOR45=BUSY NOT CLEARED BY OVERFLOW
                                        ;EROR46=DISK ERROR OCCURRED WHILE X-FERING
                                        ;EROR50=DAR DID NOT EQUAL 2 AFTER SECOND X-FER
                                        ;EROR51=WC FAILED TO OVERFLOW ON READ CONTINUE
                                        ;EROR52=CMA DID NOT EQUAL OUTBUF+2 AFTER READ CONTINUE
                                        ;
                                        ;
004344  012777  177777  174444  STAI65: MOV     #177777,@RKWC   ;SET WCI TO -1
004352  052777  000001  174434          BIS     #BIT0,@RKCS     ;SET GO TO CONTINUE
004360  105777  174430                  TSTB    @RKCS           ;TEST FOR RDY=0
004364  100012                          BPL     STAI66          ;RDY SET
004366  017767  174422  174500          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
004374                                  EROR44:
004374  012767  000056  174462  ERR56: MOV     #56,EKCOUNT     ;SET UP ERROR COUNT 56
        000057                          N=N+1
004402  004567  004130                  JSR     %5,STAER        ;REPORT BUSY NOT SET
004406  000167  177370                  JMP     STAI50
004412  005067  174456          STAI66: CLR     WRD
004416  105777  174372          INCRD:  TSTB    @RKCS           ;TST FOR RDY SET BY OVERFLOW
004422  100415                          BMI     STAI67          ;RDY SET CONTINUE
004424  005267  174444                  INC     WRD             ;WAIT FOR BUSY=0
004430  001372                          BNE     INCRD           ;GO WAIT FOR BUSY
004432  017767  174356  174434          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
004440                                  EROR45:
004440  012767  000057  174416  ERR57: MOV     #57,EKCOUNT     ;SET UP ERROR COUNT 57
        000060                          N=N+1
004446  004567  004064                  JSR     %5,STAER        ;REPORT BUSY NOT CLEARED
004452  000167  177324                  JMP     STAI50          ;RESTART ROUTINE
004456  005777  174332          STAI67: TST     @RKCS           ;DID AN ERROR OCCUR WHILE X-FERING
004462  100015                          BPL     STAI71          ;NO CONTINUE
004464  017767  174324  174402          MOV     @RKCS,WRD       ;YES! CONTENTS OF RKCS
004472  017767  174326  174376          MOV     @RKER,WRD1      ;EXT ERROR BITS
004500                                  EROR46:
004500  012767  000060  174356  ERR60: MOV     #60,EKCOUNT     ;SET UP ERROR COUNT 60
        000061                          N=N+1
004506  004567  004076                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED
004512  000167  175544                  JMP     STAI22          ;RESTART ROUTINE
004516  017767  174300  174350  STAI71: MOV     @RKDA,WRD       ;DID RKDA INCREMENT ON CONTINUE
004524  012767  000002  174344          MOV     #2,WRD1         ;WHAT RKDA SHOULD CONTAIN
004532  026767  174340  174334          CMP     WRD1,WRD        ;IS RKDA CORRECT
004540  001407                          BEQ     STAI72          ;RKDA OK
004542                                  EROR50:
004542  012767  000061  174314  ERR61: MOV     #61,EKCOUNT     ;SET UP ERROR COUNT 61
        000062                          N=N+1
004550  004567  004034                  JSR     %5,STAER1       ;REPORT RKDA INCORRECT
004554  000167  177222                  JMP     STAI50          ;RESTART ROUTINE
004560  017767  174232  174306  STAI72: MOV     @RKWC,WRD       ;FETCH WORD COUNT
004566  001407                          BEQ     STAI73          ;WORD COUNT OVERFLOWED
004570                                  EROR51:
```

```
  812  004570  012767  000062  174266  ERR62:  MOV     #62,ERCOUNT     ;SET UP ERROR COUNT 62
  813          000063                  N=N+1
  814  004576  004567  003734                  JSR     %5,STAER        ;REPORT WORD COUNT FAILED TO CLEAR
  815  004602  000167  177174                  JMP     STA156          ;RESTART ROUTINE
  816  004606  017767  174206  174260  STA173: MOV     @RKBA,WRD       ;FETCH RKBA
  817  004614  012767  011562  174254          MOV     #OUTBUF+4,WRD1  ;WHAT RKBA SHOULD EQUAL
  818  004622  026767  174250  174244          CMP     WRD1,WRD        ;IS RKBA CORRECT
  819  004630  001407                          BEQ     LPST56          ;;RKBA WAS CORRECT
  820  004632                          ERR52:
  821  004632  012767  000063  174224  ERR63:  MOV     #63,ERCOUNT     ;SET UP ERROR COUNT 63
  822          000064                  N=N+1
  823  004640  004567  003744                  JSR     %5,STAER1       ;REPORT RKBA INCORRECT
  824  004644  000167  177132                  JMP     STA156          ;RESTART ROUTINE
  825  004650  000004          LPST56: SCOPE                           ;ENTER SCOPE LOOP
  826  004652  004002                  STA156
```

```
                                        ;
                                        ;
                                        ;
                                        ;THIS ROUTINE TESTS THE ABILITY OF THE CONTROL NOT
                                        ;TO INCREMENT THE CURRENT ADDRESS REGISTER WHEN
                                        ;BIT11 OF THE RKCS REGISTER IS SET
                                        ;
                                        ;
004654  012777  000001  174132  INHCMA: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
004662  105777  174126                  TSTB    @RKCS
004666  100375                          BPL     .-4
004670  012777  177700  174120          MOV     #-100,@RKWC     ;LOAD THE WORD COUNT
004676  012777  011556  174114          MOV     #OUTBUF,@RKBA   ;LOAD THE CURRENT ADDRESS
004704  012777  004003  174102          MOV     #4003,@RKCS     ;LOAD FUNCTION REGISTER
004712  105777  174076                  TSTB    @RKCS
004716  100375                          BPL     .-4
004720  005777  174070                  TST     @RKCS           ;TEST FOR ERRORS
004724  100014                          BPL     CMAOK           ;BRANCH IF NO ERRORS
004726  017767  174062  174142          MOV     @RKCS,WRD1
004734  017767  174064  174132          MOV     @RKER,WRD
004742  012767  000064  174114  ERR64:  MOV     #64,ERCOUNT     ;SET UP ERROR COUNT 64
                000065                  N=N+1
004750  004567  003634                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED
004754  000737                          BR      INHCMA          ;LOOP ON ERROR
004756  012767  011556  174112  CMAOK:  MOV     #OUTBUF,WRD1    ;FETCH CORRECT CURRENT ADDRESS
004764  017767  174030  174102          MOV     @RKBA,WRD
004772  026767  174100  174074          CMP     WRD1,WRD        ;DID THE CUURENT ADRESS INCREMENT
005000  001406                          BEQ     LPINH           ;BRANCH RKBA DID NOT INCREMENT
005002  012767  000065  174054  ERR65:  MOV     #65,ERCOUNT     ;SET UP ERROR COUNT 65
                000066                  N=N+1
005010  004567  003574                  JSR     %5,STAER1       ;REPORT RKBA INCREMENTED
005014  000717                          BR      INHCMA          ;LOOP ON ERROR
005016  000004                  LPINH:  SCOPE
005020  004654                          INHCMA
                                        ;
                                        ;
                                        ;IN THIS STATIC TEST THE ABILITY OF THE DISK CONTROL
                                        ;TO INCREMENT THE SURFACE REGISTER WILL BE TESTED.
                                        ;
                                        ;A ONE WORD WRITE WILL BE EXECUTED
                                        ;RKDA=13
                                        ;WC!=-1
                                        ;;RKBA=OUTBUF
                                        ;
                                        ;AT THE COMPLETION OF THE WRITE
                                        ;RKDA=20
                                        ;
005022  012777  000001  173764  STA174: MOV     #BIT0,@RKCS     ;CLEAR THE DISK
005030  105777  173760                  TSTB    @RKCS
005034  100375                          BPL     .-4
005036  012777  177777  173752          MOV     #177777,@RKWC   ;SET WORD COUNT TO -1
005044  012777  011556  173746          MOV     #OUTBUF,@RKBA   ;SET UP CURRENT ADDRESS
005052  012777  000013  173742          MOV     #13,@RKDA       ;SET RKDA TO 13(LAST SECTOR ON SURFACE 0)
005060  012777  000003  173726          MOV     #3,@RKCS                ;GO WRITE ONE WORD
```

```
  881  005066  005067  174002                    CLR     WRD             ;LET'S WAIT FOR COMPLETION
  882  005072  105777  173716          LTSWT:    TSTB    @RKCS           ;TEST FOR NOT BUSY
  883  005076  100414                            BMI     STAI75          ;X-FER COMPLETE
  884  005100  005267  173770                    INC     WRD             ;+1 WAIT REG,
  885  005104  001372                            BNE     LTSWT           ;GO WAIT FOR NOT BUSY
  886  005106  017767  173702  173760            MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
  887  005114  012767  000066  173742  ERR66:    MOV     #66,ERCOUNT     ;SET UP ERROR COUNT 66
  888                  000067                    N=N+1
  889  005122  004567  003410                    JSR     %5,STAER        ;REPORT ERROR X-FER NEVER COMPLETED
  890  005126  000735                            BR      STAI74          ;RESTART ROUTINE
  891  005130  022777  000020  173664  STAI75:   CMP     #20,@RKDA       ;DID THE SURFACE INC
  892  005136  001412                            BEQ     LPST74          ;RKDA OK NO ERROR
  893  005140  017767  173656  173726            MOV     @RKDA,WRD       ;FETCH RKDA
  894  005146  012767  000067  173710  ERR67:    MOV     #67,ERCOUNT     ;SET UP ERROR COUNT 67
  895                  000070                    N=N+1
  896  005154  004567  003356                    JSR     %5,STAER        ;REPORT RKDA DID NOT INC TO 0
  897  005160  000167  177636                    JMP     STAI74          ;ERROR RE-START ROUTINE
  898  005164  000004                  LPST74:   SCOPE                   ;ENTER SCOPE LOOP
  899  005166  005022                            STAI74
```

```
 900
 901                                    ;
 902                                    ;IN THIS ROUTINE THE ABILITY OF THE CONTROL
 903                                    ;TO INCREMENT THE CYLINDER IS CHECKED
 904                                    ;
 905                                    ;ROUTINE PARAMETERS AT START
 906                                    ;RKDA=33
 907                                    ;WC1=-1
 908                                    ;;RKBA=#OUTBUF
 909                                    ;RKCS=#3
 910                                    ;
 911                                    ;PARAMETERS AT THE END OF ROUTINE
 912                                    ;RKDA=40
 913                                    ;WC1=0
 914                                    ;;RKBA=OUTBUF+2
 915                                    ;RKCS=0
 916                                    ;
 917                                    ;
 918  005170  012777  000033  173624  STA177: MOV     #33,@#RKDA      ;RKDA EQUALS THE LAST CYLINDER LAST SECTOR
 919  005176  012777  177777  173612         MOV     #177777,@#RKWC  ;WORD COUNT SET FOR ONE WORD
 920  005204  012777  011556  173606         MOV     #OUTBUF,@#RKBA  ;CURRENT ADDRESS SET UP
 921  005212  052777  000003  173574         BIS     #3,@#RKCS               ;GO WRITE
 922  005220  105777  173570                 TSTB    @#RKCS          ;IS RDY CLEARED
 923  005224  100011                         BPL     STA100          ;
 924  005226  017767  173562  173640         MOV     @#RKCS,WRD      ;FETCH CONTROL RUG
 925  005234  012767  000070  173622  ERR70: MOV     #70,ERCOUNT     ;SET UP ERROR COUNT 70
 926          000071                         N=N+1
 927  005242  004567  003270                 JSR     %5,STAER        ;REPORT BUSY NOT SET
 928  005246  000750                         BR      STA177          ;RESTART ROUTINE
 929  005250  005067  173620         STA100: CLR     WRD
 930  005254  105777  173534         INXDSK: TSTB    @#RKCS          ;IS X-FER COMPLETE
 931  005260  100415                         BMI     STA101
 932  005262  005267  173606                 INC     WRD             ;TEST READY SHOULD BE SET
 933  005266  001372                         BNE     INXDSK          ;GO CHECK AGAIN
 934  005270  017767  173520  173576         MOV     @#RKCS,WRD      ;FETCH CONTENTS OF RKCS
 935  005276  012767  000071  173560  ERR71: MOV     #71,ERCOUNT     ;SET UP ERROR COUNT 71
 936          000072                         N=N+1
 937  005304  004567  003226                 JSR     %5,STAER        ;REPORT X-FER NOT COMPLETE
 938  005310  000167  177654                 JMP     STA177          ;RESTART ROUTINE
 939  005314  022777  000040  173500 STA101: CMP     #40,@#RKDA      ;DID THE DRIVE NUMBER INC
 940  005322  001412                         BEQ     LPST77          ;YES
 941  005324  017767  173472  173542         MOV     @#RKDA,WRD      ;FETCH RKDA ITS NOT CLEAR
 942  005332  012767  000072  173524  ERR72: MOV     #72,ERCOUNT     ;SET UP ERROR COUNT 72
 943          000073                         N=N+1
 944  005340  004567  003172                 JSR     %5,STAER        ;REPORT RKDA NOT CLEAR;
 945  005344  000167  177620                 JMP     STA177          ;RESTART ROUTINE
 946  005350  000004         LPST77: SCOPE                           ;ENTER SCOPE LOOP
 947  005352  005170                 STA177
```

```
                                        ;
                                        ;
                                        ;IN THIS ROUTINE THE PROGRAM WILL GENERATE A
                                        ;OVERRUN ERROR THIS WILL BE DONE
                                        ;BY ATTEMPING A 2 BLOCK WRITE ON
                                        ;THE LAST CYLINDER AND THE LAST SECTOR ON THE
                                        ;
                                        ;
                                        ;
005354  012777  000001  173432  STA103: MOV     #BIT0,@RKCS     ;CLEAR THE DISK
005362  105777  173426                  TSTB    @RKCS
005366  100375                          BPL     .-4
005370  012777  014533  173424          MOV     #14533,@RKDA    ;SET RKDA=TO ALL ONES
005376  012777  177000  173412          MOV     #-1000,@RKWC    ;WORD COUNT=(2 BLOCKS ON HI DENSITY DRIVE)
005404  012777  011556  173406          MOV     #OUTBUF,@RKBA   ;CURRENT ADDRESS=OUTBUF
005412  012777  000003  173374          MOV     #3,@RKCS                ;GO WRITE
005420  005067  173450                  CLR     WRD
005424  105777  173364          WAFBUS: TSTB    @RKCS           ;IS RDY SET
005430  100414                          BMI     STX104          ;YES EXIT
005432  005267  173436                  INC     WRD             ;NO BUSY STILL SET
005436  001372                          BNE     WAFBUS          ;GO WAIT FOR BUSY=0
005440  017767  173350  173426          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
005446  012767  000073  173410  ERR73:  MOV     #73,ERCOUNT     ;SET UP ERROR COUNT 73
        000074                          N=N+1
005454  004567  003056                  JSR     X5,STAER        ;REPORT BUSY SET
005460  000735                          BR      STA103          ;RESTART ROUTINE
005462  032777  040000  173334  STX104: BIT     #BIT14,@RKER    ;DID THE CONTROL OVERRUN ,
005470  001014                          BNE     STA104          ;YES BRANCH
005472  017767  173316  173376          MOV     @RKCS,WRD1
005500  017767  173320  173366          MOV     @RKER,WRD
005506  012767  000074  173350  ERR74:  MOV     #74,ERCOUNT     ;SET UP ERROR COUNT 74
        000075                          N=N+1
005514  004567  003070                  JSR     X5,STAER1       ;REPORT DRIVE #DID NOT INC',
005520  000715                          BR      STA103          ;LOOP ON ERROR
005522  000004          STA104: SCOPE
005524  005354                  STA103
                                        ;
                                        ;
                                        ;
                                        ;TEST FOR NON-EXISTENT DRIVE ERROR
                                        ;THIS TEST WILL REPORT ERRORS ON A RKMRX, DRIVE SYSTEM
                                        ;8 DRIVES IS A RKMRX, DRIVE SYSTEM
005526  012777  000001  173260  NXDST:  MOV     #BIT0,@RKCS     ;CLEAR THE DRIVE
005534  105777  173254                  TSTB    @RKCS
005540  100375                          BPL     .-4
005542  012777  177777  173246          MOV     #-1,@RKWC       ;SET UP THE WORD COUNT
005550  012777  160000  173244          MOV     #160000,@RKDA   ;LOAD ADDRESS REG TO DRIVE ONE
005556  012777  000003  173230          MOV     #3,@RKCS                ;GO WRITE
005564  105777  173224                  TSTB    @RKCS
005570  100375                          BPL     .-4
005572  005777  173216                  TST     @RKCS           ;IS THE ERROR BIT SET
005576  100413                          BMI     ERNXD           ;YES THE ERROR BIT IS SET
005600  017767  173210  173270          MOV     @RKCS,WRD1
```

```
1002  005606  017767  173212  173260          MOV     @RKER,WRD
1003  005614  012767  000075  173242  ERR75:  MOV     #75,ERCOUNT     ;SET UP ERROR COUNT 75
1004          000076                  N=N+1
1005  005622  004567  002702                  JSR     %5,STAER1       ;REPORT ERROR BIT NOT SET
1006  005626  032777  000200  173170  ERNXD:  BIT     #BIT7,@RKER     ;IS NCD SET
1007  005634  001010                  BNE     STA105          ;YES: IS ERROR SET
1008  005636  017767  173152  173230          MOV     @RKCS,WRD       ;FETCH RKCS
1009  005644  012767  000076  173212  ERR76:  MOV     #76,ERCOUNT     ;SET UP ERROR COUNT 76
1010          000077                  N=N+1
1011  005652  004567  002660                  JSR     %5,STAER        ;REPORT NEED NOT SET
1012  005656  005777  173132          STA105: TST     @RKCS           ;IS ERROR FLAG SET
1013  005662  100410                  BMI     LPSX103         ;ERROR IS SET
1014  005664  017767  173124  173202          MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
1015  005672  012767  000077  173164  ERR77:  MOV     #77,ERCOUNT     ;SET UP ERROR COUNT 77
1016          000100                  N=N+1
1017  005700  004567  002632                  JSR     %5,STAER        ;REPORT ERROR NOT SET
1018  005704  000004          LPSX103:SCOPE                           ;ENTER SCOPE LOOP
1019  005706  005526                  NXDST
```

```
1020                                    ;
1021                                    ;
1022                                    ;TEST NON-EXISTENT SECTOR ERROR
1023                                    ;
1024  005710  012777  000001  173076  ST103X: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
1025  005716  105777  173072                TSTB    @RKCS
1026  005722  100375                       BPL     .-4
1027  005724  012777  000017  173070        MOV     #17,@RKDA       ;LOAD A NON-EXISTENT SECTOR
1028  005732  012777  177777  173056        MOV     #-1,@RKWC       ;LOAD THE WORD COUNT
1029  005740  012777  000003  173046        MOV     #3,@RKCS                ;GO WRITE
1030  005746  105777  173042                TSTB    @RKCS
1031  005752  100375                       BPL     .-4
1032  005754  005777  173034                TST     @RKCS           ;TEST FOR ERROR
1033  005760  100414                       BMI     LP103X          ;BRANCH IF ERROR SET
1034  005762  017767  173034  173104        MOV     @RKDA,WRD
1035  005770  017767  173020  173100        MOV     @RKCS,WRD1
1036  005776  012767  000100  173060  ERR100: MOV     #100,ERCOUNT    ;SET UP ERROR COUNT 100
1037          000101                       N=N+1
1038  006004  004567  002600                JSR     %5,STAER1       ;REPORT ERROR NOT SET
1039  006010  000737                       BR      ST103X          ;LOOP ON ERROR
1040  006012  032777  000040  173004  LP103X: BIT     #BIT5,@RKER     ;TEST FOR NXS SET
1041  006020  001014                       BNE     X103LP          ;BRANCH NXS SET
1042  006022  017767  172776  173044        MOV     @RKER,WRD
1043  006030  017767  172760  173040        MOV     @RKCS,WRD1
1044  006036  012767  000101  173020  ERR101: MOV     #101,ERCOUNT    ;SET UP ERROR COUNT 101
1045          000102                       N=N+1
1046  006044  004567  002540                JSR     %5,STAER1       ;REPORT NXS NOT SET
1047  006050  000717                       BR      ST103X          ;LOOP ON ERROR
1048  006052  000004                X103LP: SCOPE
1049  006054  005710                       ST103X
1050                                    ;
1051                                    ;
1052                                    ;TEST NON-EXISTENT CYLINDER
1053                                    ;
1054  006056  012777  000001  172730  ST104X: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
1055  006064  105777  172724                TSTB    @RKCS
1056  006070  100375                       BPL     .-4
1057  006072  012777  017740  172722        MOV     #17740,@RKDA    ;LOAD NON-EXISTENT CYLINDER
1058  006100  012777  177777  172710        MOV     #-1,@RKWC       ;LOAD WORD COUNT
1059  006106  012777  000003  172700        MOV     #3,@RKCS                ;GO WRITE
1060  006114  105777  172674                TSTB    @RKCS
1061  006120  100375                       BPL     .-4
1062  006122  005777  172666                TST     @RKCS           ;TEST FOR ERROR BIT SET
1063  006126  100414                       BMI     LP104X          ;BRANCH ERROR BIT SET
1064  006130  017767  172666  172736        MOV     @RKDA,WRD
1065  006136  017767  172652  172732        MOV     @RKCS,WRD1
1066  006144  012767  000102  172712  ERR102: MOV     #102,ERCOUNT    ;SET UP ERROR COUNT 102
1067          000103                       N=N+1
1068  006152  004567  002432                JSR     %5,STAER1       ;REPORT ERROR BIT NOT SET
1069  006156  000737                       BR      ST104X          ;LOOP ON ERROR
1070  006160  032777  000100  172636  LP104X: BIT     #BIT6,@RKER     ;TEST FOR NXC SET
1071  006166  001014                       BNE     X104LP          ;ERROR BIT SET OK
1072  006170  017767  172630  172676        MOV     @RKER,WRD
1073  006176  017767  172612  172672        MOV     @RKCS,WRD1
```

```
1074  006204  012767  000103  172652  ERR103: MOV     #103,ERCOUNT    ;SET UP ERROR COUNT 103
1075          000104                          N=N+1
1076  006212  004567  002372                  JSR     %5,STAER1       ;REPORT NXC NOT SET
1077  006216  000717                          BR      ST104X          ;LOOP ON ERROR
1078  006220  000004                  X104LP: SCOPE
1079  006222  006056                          ST104X
1080                                  ;
1081                                  ;
1082                                  ;ROUTINE TO TEST PROGRAM ERROR
1083                                  ;
1084                                  ;
1085  006224  012777  000001  172562  ST105X: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
1086  006232  105777  172556                  TSTB    @RKCS
1087  006236  100375                          BPL     .-4
1088  006240  012777  002015  172546          MOV     #2015,@RKCS     ;ISSUE SEEK WITH BIT10 SET
1089  006246  105777  172542                  TSTB    @RKCS           ;TEST FOR CONTROL READY
1090  006252  100375                          BPL     .-4             ;WAIT FOR CONTROL READY
1091  006254  005777  172534          LPX105: TST     @RKCS           ;IS THE ERROR BIT SET
1092  006260  100411                          BMI     ST106X          ;ERROR SET OK BRANCH
1093  006262  017767  172526  172604          MOV     @RKCS,WRD       ;REPORT RKCS
1094  006270  012767  000104  172566  ERR104: MOV     #104,ERCOUNT    ;SET UP ERROR COUNT 104
1095          000105                          N=N+1
1096  006276  004567  002234                  JSR     %5,STAER        ;REPORT ERROR NOT SET
1097  006302  000750                          BR      ST105X
1098  006304  032777  004000  172512  ST106X: BIT     #BIT11,@RKER    ;TEST FOR PROGRAMMING ERROR
1099  006312  001015                          BNE     LPX100          ;BIT SET OK
1100  006314  017767  172474  172554          MOV     @RKCS,WRD1
1101  006322  017767  172476  172544          MOV     @RKER,WRD
1102  006330  012767  000105  172526  ERR105: MOV     #105,ERCOUNT    ;SET UP ERROR COUNT 105
1103          000106                          N=N+1
1104  006336  004567  002246                  JSR     %5,STAER1
1105  006342  000167  177656                  JMP     ST105X          ;LOOP ON ERROR
1106  006346  000004                  LPX106: SCOPE
1107  006350  006224                          ST105X
```

```
                                        ;
                                        ;IN THIS ROUTINE THE ABILITY OF NON-EXISTENT
                                        ;MEMORY ERROR IS CHECK,
                                        ;
                                        ;TRANSFER TWO WORDS STARTING WITH THE
                                        ;LARGEST ADDRESSABLE MEMORY LOCATION IN THE
                                        ;PDP-11
                                        ;
                                        ;
006352  012777  000001  172434  NXMTSM: MOV     #BIT0,@RKCS     ;CLEAR THE DISK
006360  105777  172430                  TSTB    @RKCS
006364  100375                         BPL     .-4
006366  012777  000340  172406          MOV     #340,@PS        ;LOCK UP PROCESSOR
006374  012777  177776  172414          MOV     #-2,@RKWC       ;SET UP WORD COUNT
006402  012777  177776  172410          MOV     #177776,@RKBA   ;SET UP CURRENT ADDRESS
006410  012777  000063  172376          MOV     #63,@RKCS       ;ISSUE WRITE
006416  105777  172372                  TSTB    @RKCS           ;TEST FOR READY
006422  100375                         BPL     .-4
006424  005777  172364                  TST     @RKCS           ;TEST FOR ERROR FLAG
006430  100414                         BMI     TSTNEM          ;ERROR BIT SET OK
006432  012767  000106  172424  ERR106: MOV     #106,ERCOUNT    ;SET UP ERROR COUNT 106
        000107                          N=N+1
006440  017767  172350  172430          MOV     @RKCS,WRD1      ;FETCH FUNCTION REG
006446  017767  172352  172420          MOV     @RKER,WRD       ;FETCH ERROR REG
006454  004567  002130                  JSR     %5,STAER1       ;REPORT ERROR BIT NOT SET
006460  000734                         BR      NXMTSM          ;LOOP ON ERROR
006462  032777  040000  172324  TSTNEM: BIT     #BIT14,@RKCS
006470  001011                         BNE     HWDOK           ;BRANCH IF HARD ERROR SET
006472  012767  000107  172364  ERR107: MOV     #107,ERCOUNT    ;SET UP ERROR COUNT 107
        000110                          N=N+1
006500  017767  172310  172366          MOV     @RKCS,WRD       ;REPORT HWD NOT SET BY NEM
006506  004567  002024                  JSR     %5,STAER        ;REPORT HWD NOT SET
006512  000717                         BR      NXMTSM          ;LOOP ON ERROR
006514  032777  002000  172302  HWDOK:  BIT     #BIT10,@RKER    ;TEST FOR NEM SET
006522  001011                         BNE     CLRNEM          ;NEM SET BE X-FER OK
006524  012767  000110  172332  ERR110: MOV     #110,ERCOUNT    ;SET UP ERROR COUNT 110
        000111                          N=N+1
006532  017767  172266  172334          MOV     @RKER,WRD       ;REPORT HARD ERROR REGISTER
006540  004567  001772                  JSR     %5,STAER        ;REPORT NEM NOT SET BY X-FER
006544  000702                         BR      NXMTSM          ;LOOP ON ERROR
006546  012777  000001  172240  CLRNEM: MOV     #BIT0,@RKCS     ;CLEAR THE DISK
006554  105777  172234                  TSTB    @RKCS           ;IS THE ERROR CLEARED
006560  100375                         BPL     .-4             ;WAIT FOR READY
006562  032777  040000  172224  CKHWD:  BIT     #BIT14,@RKCS    ;IS HARD ERROR SET
006570  001412                         BEQ     CKNEM           ;HWD CLEARED BY DISK CLEAR
006572  012767  000111  172264  ERR111: MOV     #111,ERCOUNT    ;SET UP ERROR COUNT 111
        000112                          N=N+1
006600  017767  172210  172266          MOV     @RKCS,WRD       ;REPORT HWD NOT CLEARED
006606  004567  001724                  JSR     %5,STAER        ;
006612  000167  177534                  JMP     NXMTSM          ;LOOP ON ERROR
006616  032777  002000  172200  CKNEM:  BIT     #BIT10,@RKER    ;IS NEM CLEARED BY DISK CLEAR
006624  001412                         BEQ     LPNEM           ;NEM CLEARED BY DISK CLEAR
006626  012767  000112  172230  ERR112: MOV     #112,ERCOUNT    ;SET UP ERROR COUNT 112
        000113                          N=N+1
```

```
1162  006634  017767  172164  172232          MOV     @RKER,WRD       ;REPORT NEM NOT CLEARED BY DISK CLEAR
1163  006642  004567  001070                  JSR     %5,STAER
1164  006646  000167  177500                  JMP     NXMTSM          ;LOOP ON ERROR
1165  006652  000004                  LPNEM:  SCOPE                   ;ENTER SCOPE LOOP
1166  006654  006352                          NXMTSM
```

```
1167
1168                                    ;
1169                                    ;IN THIS ROUTINE THE PROGRAM WILL TEST
1170                                    ;THAT THE DISK WILL ONLY TRAP
1171                                    ;AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
1172                                    ;BY CLEARING THE DONE
1173                                    ;PROCESSOR OPERATING AT PRIORITY7
1174                                    ;
1175  006656  012777  006756  172150  STA106: MOV     #INT106,@VECTOR ;SET UP INTERRUPT VECTOR
1176  006664  012777  000340  172110          MOV     #340,@PS        ;LOCK OUT ALL INTERRUPTS
1177  006672  005077  172124                  CLR     @RKDA
1178  006676  012777  177777  172112          MOV     #177777,@RKWC   ;SET WORD COUNT TO -1
1179  006704  012777  011556  172106          MOV     #OUTBUF,@RKBA   ;LOAD CURRENT ADDRESS
1180  006712  052777  000100  172074          BIS     #100,@RKCS      ;GO  (INTERRUPT ENABLED)
1181  006720  005000                          CLR     %0              ;WAIT FOR RDY
1182  006722  005200                          INC     %0
1183  006724  001376                          BNE     .-2
1184  006726  105777  172062                  TSTB    @RKCS           ;IS CONTROL STILL NOT RDY
1185  006732  100423                          BMI     LPI06           ;NO
1186  006734  017767  172054  172132          MOV     @RKCS,WRD       ;YES!
1187  006742  012767  000113  172114  ERR113: MOV     #113,ERCOUNT    ;SET UP ERROR COUNT 113
1188          000114                          N=N+1
1189  006750  004567  001562                  JSR     %5,STAER        ;REPORT CONTROL STILL BUSY
1190  006754  000414                          BR      STA110          ;
1191                                    ;
1192                                    ;PROCESSOR SHOULD NOT TRAP TO INT106
1193                                    ;PROCESSOR PRIORITY IS LOCKED AT 7
1194                                    ;
1195  006756                            INT106:
1196  006756  012767  000114  172100  ERR114: MOV     #114,ERCOUNT    ;SET UP ERROR COUNT 114
1197          000115                          N=N+1
1198  006764  017767  172012  172102          MOV     @PS,WRD ;FETCH PROCESSOR PRIORITY
1199  006772  022626                          CMP     (6)+,(6)+
1200  006774  004567  001536                  JSR     %5,STAER        ;REPORT DISK INTERRUPTED
1201  007000  000726                          BR      STA100
1202  007002  000004                  LPI06:  SCOPE                   ;ENTER SCOPE LOOP
1203  007004  006656                          STA106
```

```
1204
1205                                    ;
1206                                    ;IN THIS ROUTINE THE PROGRAM WILL TEST
1207                                    ;THAT THE DISK WILL ONLY TRAP
1208                                    ;AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
1209                                    ;BY CLEARING THE DONE
1210                                    ;PROCESSOR OPERATING AT PRIORITY6
1211                                    ;
1212  007006  012777  007106  172020   STA110: MOV     #INT110,@VECTOR ;SET UP INTERRUPT VECTOR
1213  007014  012777  000300  171760           MOV     #300,@PS        ;LOCK OUT ALL INTERRUPTS ABOVE
1214  007022  005077  171774                   CLR     @RKDA
1215  007026  012777  177777  171762           MOV     #177777,@RKWC   ;SET WORD COUNT TO -1
1216  007034  012777  011556  171756           MOV     #OUTBUF,@RKBA   ;LOAD CURRENT ADDRESS
1217  007042  012777  000103  171744           MOV     #103,@RKCS      ;GO WRITE (INTERRUPT ENABLED)
1218  007050  005000                           CLR     %0              ;WAIT FOR NOT BUSY
1219  007052  005200                           INC     %0
1220  007054  001376                           BNE     .-2
1221  007056  105777  171732                   TSTB    @RKCS           ;IS CONTROL STILL BUSY
1222  007062  100423                           BMI     LP110           ;NO
1223  007064  017767  171724  172002           MOV     @RKCS,WRD       ;YES!
1224  007072  012767  000115  171764   ERR115: MOV     #115,ERCOUNT    ;SET UP ERROR COUNT 115
1225          000116                           N=N+1
1226  007100  004567  001432                   JSR     %5,STAER        ;REPORT CONTROL STILL BUSY
1227  007104  000414                           BR      STA112          ;
1228                                    ;
1229                                    ;PROCESSOR SHOULD NOT TRAP TO INT106
1230                                    ;PROCESSOR PRIORITY IS LOCKED AT 6
1231                                    ;
1232  007106                            IN=110:
1233  007106  012767  000116  171750   ERR116: MOV     #116,ERCOUNT    ;SET UP ERROR COUNT 116
1234          000117                           N=N+1
1235  007114  017767  171662  171752           MOV     @PS,WRD ;FETCH PROCESSOR PRIORITY
1236  007122  022626                           CMP     (6)+,(6)+
1237  007124  004567  001406                   JSR     %5,STAER        ;REPORT DISK INTERRUPTED
1238  007130  000402                           BR      STA112
1239                                    ;
1240  007132  000004                   LP110:  SCOPE                   ;ENTER SCOPE LOOP
1241  007134  007006                           STA110
1242                                    ;
1243                                    ;IN THIS ROUTINE THE PROGRAM WILL TEST
1244                                    ;THAT THE DISK WILL ONLY TRAP
1245                                    ;AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
1246                                    ;BY CLEARING THE DONE
1247                                    ;PROCESSOR OPERATING AT PRIORITY5
1248                                    ;
1249  007136  012777  007236  171670   STA112: MOV     #INT112,@VECTOR ;SET UP INTERRUPT VECTOR
1250  007144  012777  000240  171630           MOV     #240,@PS        ;LOCK OUT ALL INTERRUPTS ABOVE
1251  007152  005077  171644                   CLR     @RKDA
1252  007156  012777  177777  171632           MOV     #177777,@RKWC   ;SET WORD COUNT TO -1
1253  007164  012777  011556  171626           MOV     #OUTBUF,@RKBA   ;LOAD CURRENT ADDRESS
1254  007172  012777  000105  171614           MOV     #105,@RKCS      ;GO READ (INTERRUPT ENABLED)
1255  007200  005000                           CLR     %0              ;WAIT FOR NOT BUSY
1256  007202  005200                           INC     %0
1257  007204  001376                           BNE     .-2
```

```
1258  007206  105777  171602                TSTB    @RKCS           ;IS CONTROL STILL BUSY
1259  007212  100423                        BMI     LP112           ;NO
1260  007214  017767  171574  171652        MOV     @RKCS,WRD       ;YES!
1261  007222  012767  000117  171634  ERR117: MOV   #117,ERCOUNT    ;SET UP ERROR COUNT 117
1262          000120                        N=N+1
1263  007230  004567  001302                JSR     %5,STAER        ;REPORT CONTROL STILL BUSY
1264  007234  000414                        BR      STA114          ;
1265                                        ;
1266                                        ;PROCESSOR SHOULD NOT TRAP TO INT106
1267                                        ;PROCESSOR PRIORITY IS LOCKED AT 7
1268                                        ;
1269  007236                                IN%112:
1270  007236  012767  000120  171620  ERR120: MOV   #120,ERCOUNT    ;SET UP ERROR COUNT 120
1271          000121                        N=N+1
1272  007244  017767  171532  171622        MOV     @PS,WRD ;FETCH PROCESSOR PRIORITY
1273  007252  022626                        CMP     (6)+,(6)+
1274  007254  004567  001256                JSR     %5,STAER        ;REPORT DISK INTERRUPTED
1275  007260  000402                        BR      STA114
1276                                        ;
1277  007262  000004                        LP112:  SCOPE                   ;ENTER SCOPE LOOP
1278  007264  007136                        STA112
1279                                        ;
1280                                        ;
1281                                        ;
1282                                        ;THIS ROUTINE WILL TEST THE ABILITY OF THE DISK CONTROL
1283                                        ;TO TRAP AT BR5 WHEN THE DONE FLAG IS SET.
1284                                        ;
1285  007266  012777  000001  171520  STA114: MOV   #BIT0,@RKCS      ;CLEAR THE DISK
1286  007274  105777  171514                TSTB    @RKCS
1287  007300  100375                        BPL     .-4
1288  007302  012777  007374  171524        MOV     #INT114,@VECTOR ;SET UP DISK TRAP VECTOR
1289  007310  012777  000200  171464        MOV     #200,@PS        ;SET PROCESSOR TO PRIORITY 4
1290  007316  005077  171500                CLR     @RKDA           ;CLEAR LOW ORDER ADDRESS BITS
1291  007322  012777  177777  171466        MOV     #177777,@RKWC   ;SET WORD COUNT TO -1
1292  007330  012777  011556  171462        MOV     #OUTBUF,@RKBA   ;LOAD CURRENT ADDRESS
1293  007336  012777  000107  171450        MOV     #107,@RKCS      ;WRITE CHECK (INTERRUPT ENABLE
1294  007344  005000                        CLR     %0
1295  007346  005200                        INC     %0              ;WAIT FOR INTERRUPT TO OCCUR
1296  007350  001376                        BNE     .-2
1297  007352  012767  000121  171504  ERR121: MOV   #121,ERCOUNT    ;SET UP ERROR COUNT 121
1298          000122                        N=N+1
1299  007360  017767  171430  171506        MOV     @RKCS,WRD       ;FETCH CONTENTS OF RKCS
1300  007366  004567  001144                JSR     %5,STAER        ;REPORT INTERRUPT NO INTERRUPT
1301  007372  000403                        BR      RESTORE         ;RESTART ROUTINE
1302                                        ;
1303                                        ;
1304                                        ;
1305  007374  022626                        IN%114: CMP     (6)+,(6)+       ;TRAP OK
1306  007376  000004                        SCOPE                   ;ENTER SCOPE LOOP
1307  007400  007266                        STA114
1308                                        ;
1309                                        ;
1310                                        ;TEST RESTORE DRIVE FUNCTION
1311                                        ;
```

```
1312                                    ;
1313  007402  012777  000001  171404  RESTORE: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
1314  007410  105777  171400                   TSTB    @RKCS
1315  007414  100375                          BPL     .-4
1316  007416  005077  171360                   CLR     @PS             ;CLEAR CPU PRIORITY
1317  007422  012777  007446  171404          MOV     #RDYRES,@VECTOR ;SET UP INTERRUPT VECTOR
1318  007430  012777  000340  171400          MOV     #340,@STATUS
1319  007436  012777  000115  171350          MOV     #115,@RKCS      ;ISSUE RESTORE
1320  007444  000001                          WAIT
1321  007446  105777  171342          RDYRES: TSTB    @RKCS           ;IS THE DRIVE READY
1322  007452  100411                          BMI     TSTRES          ;BRANCH IF READY IS SET
1323  007454  012767  000122  171402  ERR122: MOV     #122,ERCOUNT    ;SET UP ERROR COUNT 122
1324          000123                          N=N+1
1325  007462  017767  171326  171404          MOV     @RKCS,WRD       ;FETCH FUNCTION REG
1326  007470  004567  001042                  JSR     %5,STAER        ;REPORT READY NOT SET
1327  007474  000742                          BR      RESTORE         ;LOOP ON ERROR
1328  007476  032777  000100  171322  TSTRES: BIT     #BIT6,@RKDS     ;TEST ARE THE HEADS MOVING
1329  007504  001414                          BEQ     RDYINT          ;BRANCH IF THE HEADS ARE MOVING
1330  007506  017767  171302  171362          MOV     @RKCS,WRD1
1331  007514  017767  171306  171354          MOV     @RKDS,WRD1
1332  007522  012767  000123  171334  ERR123: MOV     #123,ERCOUNT    ;SET UP ERROR COUNT 123
1333          000124                          N=N+1
1334  007530  004567  001054                  JSR     %5,STAER1       ;REPORT HEADS NOT MOVING
1335  007534  000722                          BR      RESTORE         ;LOOP ON ERROR
1336  007536  022626                  RDYINT: CMP     (6)+,(6)+       ;RESET STACK
1337  007540  012777  007554  171266          MOV     #THREST,@VECTOR ;SETUP VECTOR ADDRESS
1338  007546  005077  171230                  CLR     @PS             ;CLEAR STATUS
1339  007552  000001                          WAIT
1340  007554  022626                  THREST: CMP     (6)+,(6)+       ;RESET THE STACK
1341  007556  032777  020000  171230          BIT     #BIT13,@RKCS    ;TEST FOR SEARCH COMPLETE
1342  007564  001773                          BEQ     THREST          ;WAIT FOR DRIVE READY
1343  007566  005777  171222                  TST     @RKCS           ;IS THERE AN ERROR
1344  007572  100014                          BPL     LPREST          ;NO ERROR RESTORE OK
1345  007574  017767  171214  171274          MOV     @RKCS,WRD1
1346  007602  017767  171220  171264          MOV     @RKDS,WRD
1347  007610  012767  000124  171246  ERR124: MOV     #124,ERCOUNT    ;SET UP ERROR COUNT 124
1348          000125                          N=N+1
1349  007616  004567  000766                  JSR     %5,STAER1       ;REPORT ERROR WHILE RESTORING
1350  007622  000667                          BR      RESTORE         ;LOOP ON ERROR
1351  007624  000004                  LPREST: SCOPE
1352  007626  007402                          RESTORE
1353                                  ;* * * ADDRESS TEST 1 * * *
1354                                  ;
1355                                  ;
1356                                  ;READ SECTOR FORMAT AND VERIFTY THE CYLINDER
1357                                  ;ADDRESS IS CORRECT
1358                                  ;
1359                                  ;NOTE: DATA IS NOT CHECKED IN TEST
1360                                  ;
1361                                  ;
1362  007630  012706  001000          AD11:   MOV     #1000,%6        ;SET UP THE STACK
1363  007634  005067  171210                  CLR     DMA
1364  007640  016777  171204  171154  RDFMT: MOV     DMA,@RKDA       ;LOAD THE DRIVE ADDRESS
1365  007646  012777  177752  171142          MOV     #-26,@RKWC              ;LOAD WORD COUNT
```

```
1366  007654  012777  011556  171136          MOV     #OUTBUF,@RKBA   ;LOAD CURRENT ADDRESS
1367  007662  012777  002005  171124          MOV     #2005,@RKCS     ;LOAD READ FORMAT
1368  007670  105777  171120                  TSTB    @RKCS           ;TEST FOR CONTROL READY
1369  007674  100375                          BPL     .-4
1370  007676  005777  171112                  TST     @RKCS           ;TEST FOR ERROR
1371  007702  100014                          BPL     NOFMT           ;BRANCH IF NO ERROR
1372  007704  017767  171104  171164          MOV     @RKCS,WRD1
1373  007712  017767  171104  171154          MOV     @RKDA,WRD
1374  007720  012767  000125  171136  ERR125:  MOV     #125,ERCOUNT    ;SET UP ERROR COUNT 125
1375          000126                          N=N+1
1376  007726  004567  000656                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED
1377  007732  000742                          BR      RQFMT           ;LOOP ON ERROR
1378  007734  005067  171126  NOFMT:  CLR     SAVE
1379  007740  016767  171104  171130          MOV     DMA,WRD1
1380  007746  012701  011556                  MOV     #OUTBUF,%1
1381  007752  026721  171120  CMPNEX:  CMP     WRD1,(1)+       ;IS THE FORMAT CORRECT
1382  007756  001411                          BEQ     CHKNEX          ;FORMAT CORRECT BRANCH
1383  007760  016767  171102  171106          MOV     SAVE,WRD
1384  007766  012767  000126  171070  ERR126:  MOV     #126,ERCOUNT    ;SET UP ERROR COUNT 126
1385          000127                          N=N+1
1386  007774  004567  000610                  JSR     %5,STAER1       ;REPORT SECTOR IN ERRROR
1387  010000  000717                          BR      RQFMT           ;CHECK NEXT SECTOR
1388  010002  005267  171060  CHKNEX:  INC     SAVE
1389  010006  022767  000026  171052          CMP     #26,SAVE        ;HAVE ALL SECTORS BEEN CHECKED
1390  010014  001356                          BNE     CMPNEX          ;NO CHECK NEXT ONE
1391  010016  022767  014500  171024          CMP     #14500,DMA      ;IS THIS THE LAST CYLINDER
1392  010024  001404                          BEQ     LPADT           ;YES EXIT
1393  010026  062767  000040  171014          ADD     #40,DMA         ;SET UP NEXT CYLINDER
1394  010034  000701                          BR      RQFMT
1395  010036  032777  004000  170734  LPADT:  BIT     #BIT11,@SWR     ;LOOP ON TEST
1396  010044  001402                          BEQ     .+6
1397  010046  000167  177556                  JMP     ADT1            ;LOOP ON THE TEST
1398                                  ;
1399                                  ;
1400                                  ;
1401                                  ;
1402                                  ;
1403
1404  010052  104001                          EMT+1                   ;REPORT END
1405  010054  011550                          END
1406  010056  013700  000042                  MOV     @#42,%0         ;GET MONITOR ADDRESS
1407  010062  001404                          BEQ     LXIT            ;SKIP IF NOT MONITOR LOAD
1408  010064  004710                  LOGICAL:  JSR     7,(0)           ;GO TO MONITOR
1409  010066  000240                          NOP
1410  010070  000240                          NOP
1411  010072  000240                          NOP
1412  010074  000167  171132          LXIT:   JMP     STAI1           ;RESTART TEST
1413                                  ;
1414                                  ;SCOPE LOOP ROUTINE
1415                                  ;IF BIT 11 SET LOOP ON TEST
1416                                  ;
1417  010100  032777  004000  170672  LOOP:   BIT     #BIT11,@SWR     ;TST FOR BIT 11
1418  010106  001402                          BEQ     .+6             ;BIT 11 NOT SET
1419  010110  013646                          MOV     @(6)+,-(6)
```

```
1420  010112  000002                          RTI                     ;BIT 11 SET! LOOP ON TEST
1421  010114  005767  170752                  TST     PASS            ;TEST TO SET UP PASS COUNT
1422  010120  001003                          BNE     .+10            ;PASS COUNT SET
1423  010122  012767  000025  170742          MOV     #25,PASS        ;SET UP PASS COUNT
1424  010130  005367  170736                  DEC     PASS            ;SUB, #1 EACH PASS
1425  010134  001402                          BEQ     .+6
1426  010136  013646                          MOV     @(6)+,-(6)
1427  010140  000002                          RTI                     ;LOOP ON TEST
1428  010142  062716  000002                  ADD     #2,(6)          ;INDEX POINTER FOR NEXT TEST
1429  010146  000002                          RTI                     ;EXIT TO NEXT TEST
```

```
1430
1431                                    ;
1432                                    ;
1433                                    ;
1434                                    ;
1435                                    ;
1436                                    ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
1437                                    ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
1438                                    ;BYTE OF THE CONTROL REGISTER.
1439                                    ;
1440  010150  016705  170654           DISK1   MOV     RKDB,R5         ;SET UP TO LOAD DISK REG
1441  010154  016745  170666                   MOV     TRACK,2(5)      ;LOAD TRACK NUMBER
1442  010160  016745  170664                   MOV     DMA,4(5)        ;LOAD WORD ADDRESS
1443  010164  016745  170664                   MOV     BUF,4(5)        ;SET UP CURRENT ADDRESS
1444  010170  016745  170650                   MOV     WRDCT,2(5)      ;LOAD WORD COUNT
1445  010174  005115                           COM     (5)             ;SET UP TWO'S COMPLEMENT
1446  010176  005215                           INC     (5)
1447  010200  011604                           MOV     (6),R4
1448  010202  014467  170666                   MOV     -(4),WRD        ;
1449  010206  042767  177600  170660           BIC     #177600,WRD     ;RKMRSK FUNCTION BITS
1450  010214  016745  170654                   MOV     WRD,-(5)        ;LOAD FUNCTION REG.
1451  010220  000002                           RTI                     ;RETURN FROM TRAP
1452                                    ;
1453                                    ;
1454                                    ;
1455                                    ;
1456                                    ;
1457                                    ;
1458                                    ;
1459  010222  012767  010246  167574   DOWN1   MOV     #UP,24          ;SET UP POWER FAIL VECTOR
1460  010230  012777  000001  170556           MOV     #BIT0,@RKCS     ;ABORT DISK
1461  010236  105777  170552                   TSTB    @RKCS
1462  010242  100375                           BPL     .-4
1463  010244  000000                           HALT
1464                                    ;
1465                                    ;
1466  010246  012767  010222  167550   UP1     MOV     #DOWN,24        ;SET UP FOR NEXT POWER FAIL
1467  010254  105777  170546                   TSTB    @RKCS           ;TEST FOR DRIVE READY
1468  010260  100375                           BPL     .-4
1469  010262  104001                           EMT+1
1470  010264  011460                           PWRF
1471  010266  000167  170610                   JMP     START
1472                                    ;
1473                                    ;
1474                                    ;
1475                                    ;
1476                                    ;
1477
1478                                    ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
1479                                    ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
1480                                    ;
1481                                    ;WORD COUNT REGISTER
1482  010272  017777  170502  170516   SELWC1  MOV     @SWR,@RKWC              ;MOV SWR INTO WORD COUNT REG
1483  010300  000774                           BR      SELWC
```

```
1484                                    ;
1485                                    ;CURRENT ADDRESS REGISTER
1486  010302  017777  170472  170510  SELCMA: MOV     @SWR,@RKBA      ;MOV SWR INTO CURRENT ADDR REG
1487  010310  000774                          BR      SELCMA
1488                                    ;
1489                                    ;DISK ADDRESS REGISTER
1490  010312  017777  170462  170502  SELDAR: MOV     @SWR,@RKDA      ;MOV SWR INTO DISK ADDR REG
1491  010320  000774                          BR      SELDAR
1492                                    ;
1493                                    ;
1494                                    ;DRIVE CONTROL STATUS REGISTER
1495  010322  012777  000340  170452  SELDCS: MOV     #340,@PS        ;LOCK UP INTERRUPTS
1496  010330  012777  177777  170460          MOV     #177777,@RKWC   ;SET WORD COUNT =1 WORD
1497  010336  012777  011556  170454          MOV     #OUTBUF,@RKBA   ;SET UP CURRENT ADDRESS
1498  010344  017777  170430  170442          MOV     @SWR,@RKCS      ;MOV SWR INTO CONTROL REG
1499  010352  032777  000001  170434          BIT     #BIT0,@RKCS     ;IS FUNCTION BITS SET
1500  010360  001760                          BEQ     SELDCS          ;FUNCTION BITS NOT SET
1501  010362  105777  170426          DKBUSY: TSTB    @RKCS           ;TEST FOR DRIVE READY
1502  010366  100375                          BPL     DKBUSY          ;DRIVE STILL NOT READY
1503  010370  000754                          BR      SELDCS          ;DRIVE NOT BUSY SELECT NEW CR
1504                                    ;
1505                                    ;
1506                                    ;
1507                                    ;
1508                                    ;
1509                                    ;
1510                                    ;
```

```
1511
1512                                    ;
1513                                    ;THIS ROUTINE ENABLES THE OPERATOR TO SELECT A TRACK STATICLY
1514                                    ;THE ROUTINE DOES A ONE WORD READ TO SELECT THE TRACK
1515                                    ;THE OPERATOR MAY CHANGE THE SWITCH REGISTER AT ANY TIME
1516                                    ;SET DRIVE AND CYLINDER NO, IN SWR11-0
1517                                    ;
1518  010372  012777  000001  170414  TKSEL: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
1519  010400  105777  170410          TSTB    @RKCS
1520  010404  100375                  BPL     .-4
1521  010406  017767  170366  170462        MOV     @SWR,WRD1       ;FETCH SWR
1522  010414  016767  170456  170452        MOV     WRD1,WRD
1523  010422  042767  174000  170444        BIC     #174000,WRD     ;MASK DISK AND TRACK NO,
1524  010430  000241                  CLC
1525  010432  006167  170436          ROL     WRD
1526  010436  006167  170432          ROL     WRD
1527  010442  006167  170426          ROL     WRD
1528  010446  006167  170422          ROL     WRD
1529  010452  006167  170416          ROL     WRD
1530  010456  016777  170412  170336        MOV     WRD,@RKDA       ;DRIVE ADDRESS REG LOADED
1531  010464  012777  011566  170326        MOV     #INBUF,@RKBA    ;LOAD CURRENT ADDRESS
1532  010472  012777  177777  170316        MOV     #177777,@RKWC   ;LOAD WORD COUNT
1533  010500  052777  000005  170306        BIS     #5,@RKCS                ;GO AND READ
1534  010506  032777  000100  170312  CTBUSY: BIT     #BIT6,@RKDS     ;ARE THE HEADS STILL MOVING
1535  010514  001774                  BEQ     .-6
1536  010516  105777  170272          TSTB    @RKCS           ;TEST FOR CONTROL READY
1537  010522  100371                  BPL     CTBUSY          ;WAIT FOR CONTROL READY
1538  010524  026777  170346  170246  SRCHG:  CMP     WRD1,@SWR
1539  010532  001317                  BNE     TKSEL           ;SWR HAS CHANGED
1540  010534  000773                  BR      SRCHG           ;SWR HAS NOT CHANGED
1541                                    ;
1542                                    ;
1543                                    ;
1544                                    ;
1545                                    ;
1546                                    ;
```

```
1547
1548                                    ;
1549                                    ;ROUTINE TO REPORT ERROR COUNT AND CONTENTS OF ONE REGISTER
1550                                    ;
1551  010536  004567  000404           STAER1  JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1552  010542  001074                           WRD                     ;DATA TO BE CONVERTED
1553  010544  011530                           MES6                    ;ADDRESS OF MESSAGE
1554  010546  000006                           6
1555  010550  004567  000372                   JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1556  010554  001064                           ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
1557  010556  011414                           HED5                    ;ADDRESS OF MESSAGE
1558  010560  000003                           3
1559  010562  104000                           EMT     +0              ;REPORT MESSAGE
1560  010564  011412                           HED5A
1561  010566  011414                           HED5
1562  010570  011530                           MES6
1563  010572  177777                           -1
1564  010574  032777  002000  170176           BIT     #BIT10,@SWR
1565  010602  001401                           BEQ     .+4
1566  010604  000000                           HALT
1567  010606  000205                           RTS     X5              ;EXIT ROUTINE
1568                                    ;
1569                                    ;ROUTINE TO REPORT ERROR COUNT AND THE CONTENTS OF TWO REGISTERS
1570                                    ;
1571                                    ;
1572                                    ;
1573  010610  004567  000332           STAER11 JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1574  010614  001074                           WRD                     ;DATA TO BE CONVERTED
1575  010616  011530                           MES6                    ;ADDRESS OF MESSAGE
1576  010620  000006                           6
1577  010622  004567  000320                   JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1578  010626  001076                           WRD1                    ;DATA TO BE CONVERT
1579  010630  011435                           MES5                    ;ADDRESS OF MESSAGE
1580  010632  000006                           6
1581  010634  004567  000306                   JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1582  010640  001064                           ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
1583  010642  011414                           HED5                    ;ADDRESS OF MESSAGE
1584  010644  000003                           3
1585  010646  104000                           EMT     +0              ;REPORT MESSAGE
1586  010650  011412                           HED5A
1587  010652  011414                           HED5
1588  010654  011435                           MES5
1589  010656  011530                           MES6
1590  010660  177777                           -1
1591  010662  032777  002000  170110           BIT     #BIT10,@SWR
1592  010670  001401                           BEQ     .+4
1593  010672  000000                           HALT
1594  010674  000205                           RTS     X5              ;EXIT ROUTINE
1595                                    ;
1596                                    ;
1597                                    ;ROUTINE TO DECODE EMT CALLS
1598                                    ;EMT-1=TYPE ONE LINE OF TEXT
1599                                    ;EMT-0=TYPE A SERIES OF LINES
1600  010676  011600                   EMTRP1  MOV     (0),X0
```

```
1601  010700  022740  104001          CMP     #EMT+1,-(0)     ;WAS THE CALL EMT+1
1602  010704  001103                  BNE     TYPS            ;NO! TYPE A SERIES OF LINES OF TEXT
1603  010706  000400                  BR      TYP             ;YES TYPE ONE LINE OF TEXT
```

```
1604
1605                                    ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER,
1606  010710  011600                    TYP:    MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1607  010712  062716  000002                    ADD     #2,@%6          ;SET UP EXIT,
1608  010716  011000                            MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0,
1609  010720  112067  000164            TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
1610  010724  122767  000100  000156            CMPB    #100,TYPDAT     ;CHECK FOR "@" CHARACTER
1611  010732  001005                            BNE     TYPC            ;BRANCH IF NOT "@",
1612  010734  005067  000150                    CLR     TYPDAT          ;IF END OUTPUT
1613  010740  004767  000030                    JSR     %7,TYPD         ;NULL TO CLEAR BUFFER
1614  010744  000002                            RTI                     ;TERMINATOR CHAR, DONE, EXIT,
1615  010746  122767  000045  000134    TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR "%",
1616  010754  001442                            BEQ     TYPF            ;BRANCH IF "%",
1617  010756  122767  000042  000124            CMPB    #42,TYPDAT      ;NOT "%", CHECK FOR "#",
1618  010764  001443                            BEQ     TYPG            ;BRANCH IF "#"
1619  010766  004767  000002                    JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
1620  010772  000752                            BR      TYPA
1621  010774  032777  040000  167776    TYPD:   BIT     #BIT14,@SWR
1622  011002  001026                            BNE     TYEXIT
1623  011004  116777  000100  167772            MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
1624  011012  105777  167772                    TSTB    @TPS            ;WAIT FOR DONE FLAG,
1625  011016  100375                            BPL     .-4
1626  011020  122767  000015  000062            CMPB    #15,TYPDAT      ;CHECK FOR CR
1627  011026  001003                            BNE     1$              ;NO-SKIP
1628  011030  012767  000011  000054            MOV     #9.,NULL        ;SET NULL COUNTER
1629  011036  005767  000050            1$:     TST     NULL            ;TEST COUNTER
1630  011042  001406                            BEQ     TYEXIT          ;ZERO-EXIT
1631  011044  005367  000042                    DEC     NULL            ;DECREMENT
1632  011050  112767  000000  000032            MOVB    #0,TYPDAT       ;ZERO OUTPUT
1633  011056  000746                            BR      TYPD            ;OUTPUT NULL
1634  011060  000207                    TYEXIT: RTS     %7              ;EXIT
1635  011062  112767  000015  000020    TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1636  011070  004767  177700                    JSR     %7,TYPD         ;GO TYPE CHAR,
1637  011074  112767  000012  000006    TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT,
1638  011102  004767  177666                    JSR     %7,TYPD         ;GO TYPE CHAR,
1639  011106  000704                            BR      TYPA
1640  011110  000000                    TYPDAT: 0
1641  011112  000000                    NULL:   0
1642                                    ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
1643  011114  011600                    TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1644  011116  062716  000002                    ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
1645  011122  011067  000014                    MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
1646  011126  022767  177777  000006            CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
1647  011134  001001                            BNE     TYPSA           ;BRANCH IF NOT TERMINATOR,
1648  011136  000002                            RTI                     ;TERMINATOR, EXIT
1649  011140  104001                    TYPSA:  EMT     +1              ;CALL ON TYP SUB TO TYPE MESSAGE
1650  011142  000000                    TYPSB:  0                       ;ADDRESS OF MESSAGE GOES HERE
1651  011144  000763                            BR      TYPS            ;GO PROCESS NEXT MESSAGE
1652                                    ;
1653                                    ;
1654                                    ;
1655                                    ;
1656                                    ;OCTAL TO ASCII CONVERT ROUTINE
1657                                    ;
```

```
                                ;ENTER ROUTINE AS FOLLOWS
                                ;JSR%5,CONV
                                ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED
                                ;ADDR BYTE=LSB OF WHERE ASCII IS GOING
                                ;ASCII#=THE NUMBER OF ASCII CHAR, TO BE CONVERTED
                                ;
                                ;
011146  013567  000054          CONV:   MOV     @(5)+,ACNVX     ;VALUE OF # TO BE CONVERTED
011152  012501                          MOV     (5)+,%1         ;ASCII ADDR
011154  012502                          MOV     (5)+,%2         ;# OF ASCII CHAR
011156  060201                          ADD     %2,%1
```

```
1669
1670  011160  016703  000042          ACVN:   MOV     ACNVX,R3
1671  011164  042703  177770                  BIC     #177770,R3      ;ISOLATE LEAST SIGNIFICANT OCTAL#
1672  011170  062703  000060                  ADD     #60,R3          ;SET UP ASCII#
1673  011174  110341                          MOVB    R3,-(R1)        ;STORE ASCII CHAR
1674  011176  042767  000007  000022          BIC     #7,ACNVX
1675  011204  006067  000016                  ROR     ACNVX           ;ROTATE OCTAL#
1676  011210  006067  000012                  ROR     ACNVX
1677  011214  006067  000006                  ROR     ACNVX
1678  011220  005302                          DEC     R2              ;-1 FROM ASCII CHAR COUNT
1679  011222  001356                          BNE     ACVN
1680  011224  000205                          RTS     R5              ;EXIT # CONVERTED
1681  011226  000000          ACNVX:  0                               ;WRD REGISTER
1682                                  ;
1683                                  ;
1684                                  ;
1685                                  ;
1686                                  ;ROUTINE TO SET ACTION ENABLE ON MA/MF PARITY MEMORIES
1687                                  ;CALL JSR, PC, MAMF
1688
1689          172100                  PARCSR=172100
1690          000114                  PARVEC=114
1691          000004                  ERRVEC=4
1692          000006                  SP=%6
1693
1694  011230  012737  011322  000114  MAMF:   MOV     #PARSRV,@#PARVEC        ;SET PARITY INTERRUPT VECTOR
1695  011236  012737  000340  000116          MOV     #340,@#PARVEC+2 ;AND PRIORITY LEVEL 7 ON INTERRUPT
1696  011244  013746  000004                  MOV     @#ERRVEC,-(SP)  ;SAVE CURRENT ERROR VECTOR
1697  011250  013746  000006                  MOV     @#ERRVEC+2,-(SP) ;AND PRIORITY LEVEL
1698  011254  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC
1699  011262  012737  000002  000006          MOV     #RTI,@#ERRVEC+2
1700  011270  012700  172100                  MOV     #PARCSR,R0      ;GET FIRST CSR ADDRESS
1701  011274  012702  000001                  MOV     #1,R2
1702  011300  012720  000001          1$:     MOV     #1,(R0)+        ;SET ACTION ENABLE IF AVAILABLE
1703  011304  006302                          ASL     R2              ;SHIFT AVAILABILITY INDICATOR
1704  011306  103374                          BCC     1$
1705  011310  012637  000006                  MOV     (SP)+,@#ERRVEC+2 ;RESTORE ERROR VECTOR
1706  011314  012637  000004                  MOV     (SP)+,@#ERRVEC  ;PRIORITY LEVEL AND INTERRUPT VECTOR
1707  011320  000207                          RTS     R7
1708                                  ;PARITY ERROR SERVICE ROUTINE
1709                                  ;WHEN A PARITY ERROR IS DETECTED THE ROUTINE SCANS
1710                                  ;MEMORY FOR THE PARITY ERROR WHEN THE ERROR
1711                                  ;IS DETECTED THE PROGRAM HALTS WITH THE ADDRESS
1712                                  ;CAUSING THE ERROR IN R0
1713                                  ;TO CONTINUE PRESS CONTINUE
1714
1715  011322  104001          PARSRV: EMT+1
1716  011324  011503                  PARERR
1717  011326  012737  011354  000114          MOV     #2$,@#PARVEC    ;REPOSITION PARITY ERROR INT,
1718  011334  012737  011402  000004          MOV     #4$,@#ERRVEC    ;SET TIME OUT TRAP
1719  011342  005037  000006                  CLR     @#ERRVEC+2
1720  011346  005000                          CLR     R0
1721  011350  005720          1$:     TST     (R0)+           ;SCAN MEMORY
1722  011352  000776                  BR      1$
```

```
1723  011354  000000                  2$:     HALT                    ;PARITY ERROR-ADDRESS
1724                                                                  ;CAUSING ERROR IS IN REGISTER 0
1725  011356  000005                  3$:     RESET
1726  011360  012737  011322  000114          MOV     #PARSKV,@#PARVEC        ;RESTORE PARITY VECTOR
1727  011366  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;RESTORE TIME OUT HALT
1728  011374  004767  177630                  JSR     %7,MAMF
1729  011400  000002                          RTI
1730  011402  000000                  4$:     HALT                    ;ERROR-PARITY ERROR NOT DETECTED ON SCAN
1731  011404  000764                          BR      3$              ;4(SP) CONTAINS PC WHERE
1732                                                                  ;PARITY ERROR WAS ORIGINALLY DETECTED
1733
```

```
1734
1735  011406  000000                  TEXBUF: 0
1736  011410  000000                  TSTCH:  0
1737                                  ;
1738                                  ;
1739                                  ;
1740                                  ;
1741                                  ;
1742                                  ;ERROR MESSAGE HEADERS
1743                                  ;
1744
1745                                          .EVEN
1746  011412  040045                  HED5A:  .ASCII  /%*/
1747  011414  020040  020040  051105  HED5:   .ASCII  /    ERROR COUNT */
1748  011422  047522  020122  047503
1749  011430  047125  020124     100
1750                                  ;
1751                                  ;
1752                                  ;MESSAGE TRAILERS
1753                                  ;
1754                                  ;
1755                                  ;
1756                                  ;
1757  011435     040  020040  020040  MES5:   .ASCII  /       GOOD DATA  */
1758  011442  020040  047507  042117
1759  011450  042040  052101  020101
1760  011456  040040
1761                                  ;
1762                                  ;
1763  011460  020045  047520  042527  PWRF:   .ASCII  /% POWER HAS FAILED*/
1764  011466  020122  040510  020123
1765  011474  040506  046111  042105
1766  011502     100
1767                                  ;
1768  011503     045  042515  047515  PARERR: .ASCII  /%MEMORY PARITY ERROR*/
1769  011510  054522  050040  051101
1770  011516  052111  020131  051105
1771  011524  047522  040122
1772  011530  020040  020040  020040  MES6:   .ASCII  /       BAD DATA*/
1773  011536  041040  042101  042040
1774  011544  052101  040101
1775                                  ;
1776                                  ;
1777                                  ;
1778  011550  042445  042116     100  END:    .ASCII  /%END*/
```

```
1779
1780            011556                          .EVEN
1781                                    ;
1782                                    ;
1783                                    ;
1784  011556   000000   000000   000000  OUTBUFI  .WORD   0,0,0,0
1785  011564   000000
1786                                    ;
1787                                    ;
1788  011566   000000   000000   000000  INBUFI   .WORD   0,0,0,0
1789  011574   000000
1790                                    ;
1791                                    ;
1792                                    ;
1793                                    ;
1794            000001                          .END
```

```
ACNVX   011226    ACVN    011160    ADT1    007630    BIT0  = 000001
BIT1  = 000002    BIT10 = 002000    BIT11 = 004000    BIT12 = 010000
BIT13 = 020000    BIT14 = 040000    BIT15 = 100000    BIT2  = 000004
BIT3  = 000010    BIT4  = 000020    BIT5  = 000040    BIT6  = 000100
BIT7  = 000200    BIT8  = 000400    BIT9  = 001000    BUF     001054
CHKNEX  010002    CKHWD   006562    CKNEM   006616    CLRNEM  006546
CMAOK   004756    CMPNEX  007752    CONV    011446    CTBUSY  010506
DISK    010150    DKBUSY  010362    DMA     001050    DOWN    010222
EMTRP   010676    END     011550    ERCOUN  001064    ERNXD   005026
EROR1   002342    EROR10  002654    EROR11  002720    EROR12  002760
EROR14  003022    EROR15  003050    EROR16  003112    EROR17  003214
EROR2   002416    EROR20  003270    EROR21  003322    EROR23  003364
EROR24  003412    EROR25  003454    EROR26  003522    EROR27  003566
EROR3   002454    EROR30  003626    EROR32  003670    EROR33  003710
EROR34  003760    EROR35  004062    EROR36  004136    EROR37  004174
EROR41  004236    EROR42  004264    EROR43  004320    EROR44  004374
EROR45  004440    EROR46  004500    EROR5   002516    EROR50  004542
EROR51  004570    EROR52  004632    EROR6   002544    EROR7   002606
ERRVEC = 000004   ERR0    001252    ERR1    001272    ERR10   001570
ERR100  005776    ERR101  000036    ERR102  006144    ERR103  006204
ERR104  006270    ERR105  006330    ERR106  006432    ERR107  006472
ERR11   001654    ERR110  006524    ERR111  006572    ERR112  006626
ERR113  006742    ERR114  006756    ERR115  007072    ERR116  007106
ERR117  007222    ERR12   001724    ERR120  007256    ERR121  007352
ERR122  007454    ERR123  007522    ERR124  007610    ERR125  007720
ERR126  007766    ERR13   002000    ERR14   002050    ERR15   002120
ERR16   002170    ERR17   002244    ERR2    001322    ERR20   002342
ERR21   002416    ERR22   002454    ERR23   002516    ERR24   002544
ERR25   002606    ERR26   002654    ERR27   002720    ERR3    001352
ERR30   002760    ERR31   003022    ERR32   003050    ERR33   003112
ERR34   003214    ERR35   003270    ERR36   003322    ERR37   003364
ERR4    001402    ERR40   003412    ERR41   003454    ERR42   003522
ERR43   003566    ERR44   003626    ERR45   003670    ERR46   003710
ERR47   003760    ERR5    001432    ERR50   004062    ERR51   004136
ERR52   004174    ERR53   004236    ERR54   004264    ERR55   004320
ERR56   004374    ERR57   004440    ERR6    001462    ERR60   004500
ERR61   004542    ERR62   004570    ERR63   004632    ERR64   004742
ERR65   005002    ERR66   005114    ERR67   005146    ERR7    001512
ERR70   005234    ERR71   005276    ERR72   005332    ERR73   005440
ERR74   005506    ERR75   005614    ERR76   005644    ERR77   005672
FLAG    001040    HED5    011414    HED5A   011412    HWDOK   006514
INBUF   011566    INCBUS  002676    INCRD   004416    INCWAT  002362
INHCMA  004654    INT106  006756    INT110  007106    INT112  007230
INT114  007374    INXDSK  005254    LOGICA  010004    LOOP    010100
LPADT   010036    LPINH   005016    LPNEM   006652    LPREST  007024
LPST1   001304    LPST10  001602    LPST11  001606    LPST12  001730
LPST13  002012    LPST14  002062    LPST15  002132    LPST16  002202
LPST17  002256    LPST2   001334    LPST22  003130    LPST3   001364
LPST4   001414    LPST40  003776    LPST5   001444    LPST56  004050
LPST6   001474    LPST7   001524    LPST74  005164    LPST77  005350
LPSX10  005704    LPX105  006254    LPX106  006346    LPX03X  006012
LP104X  006160    LP106   007002    LP110   007132    LP112   007262
LTSWT   005072    LXIT    010074    MAMF    011230    MES5    011434
MES6    011530    N     = 000127    NOFMT   007734    NULL    011112
```

```
NXDST   005526    NXMISM  006352    OUTBUF  011556    PARCSR = 172100
PARERR  011503    PARSRV  011322    PARVEC = 000114    PASS    001072
PATNU   001052    PS      001002    PWRF    011460    RANNU   001042
RDFMT   007640    RDWAT   004102    RDYINT  007536    RDYRES  007446
READ  = 104405    RESTOR  007402    RKBA    001020    RKCS    001014
RKDA    001022    RKDB    001030    RKDS    001026    RKER    001024
RKMR    001032    RKWC    001016    SAVE    001066    SAV1    001070
SCOPE = 000004    SELCMA  010302    SELDAR  010312    SELDCS  010322
SELWC   010272    SP    =X000006    SRCHG   010524    SRDCT   001062
STAER   010536    STAER1  010610    STA11   001232    STA110  001530
STA111  001606    STA112  001672    STA113  001742    STA114  002010
STA115  002066    STA116  002136    STA117  002206    STA12   001310
STA122  002262    STA123  002356    STA124  002432    STA126  002472
STA127  002534    STA13   001340    STA130  002562    STA131  002624
STA132  002672    STA133  002736    STA135  002776    STA136  003040
STA137  003066    STA14   001370    STA140  003134    STA141  003230
STA142  003304    STA144  003340    STA145  003402    STA146  003430
STA147  003472    STA15   001420    STA150  003540    STA151  003604
STA153  003644    STA154  003706    STA155  003734    STA156  004002
STA157  004076    STA16   001450    STA160  004152    STA162  004212
STA163  004254    STA164  004302    STA165  004344    STA166  004412
STA167  004456    STA17   001500    STA171  004516    STA172  004560
STA173  004606    STA174  005022    STA175  005130    STA177  005170
START   001102    STATUS  001036    STA100  005250    STA101  005314
STA103  005354    STA104  005522    STA105  005656    STA106  006656
STA110  007006    STA112  007136    STA114  007266    STX104  005462
ST103X  005710    ST104X  006056    ST105X  006224    ST106X  006304
SWR     001000    TDMA    001060    TEXBUF  011406    THREST  007554
TKB     001006    TKS     001012    TKSEL   010372    TPB     001004
TPS     001010    TRACK   001046    TSTCH   011410    TSTNEM  006462
TSTRES  007476    TWRDCT  001056    TYEXIT  011060    TYP     010710
TYPA    010720    TYPC    010746    TYPD    010774    TYPDAT  011110
TYPF    011062    TYPG    011074    TYPS    011114    TYPSA   011140
TYPSB   011142    UP      010246    VECTOR  001034    WAFBUS  005424
WCBUSI  003544    WCWAT   003234    WRCHEC = 104407    WRD     001074
WRDCT   001044    WRD1    001076    WRD2    001100    WRITE = 104403
XSTA11  001264    X103LP  006052    X104LP  006220    .     = 011576
```

ERRORS DETECTED: 0

```
*DZRKAA:DZRKAA/SOL=DZRKAA:BIC
RUN-TIME:  9 18 0 SECONDS
CORE USED:  4K
```

digital

MAINDEC CHANGE NOTICE

MCN 901
~~36 MCN~~

CHANGE NO.
11-DZRKB-A - 1
Sheet 1 of 1

AUTHOR	PROGRAM DATE	PRODUCT LINE	MAINDEC NUMBER
J. Hittell DATE EXT. 5/11/73 2173	MARCH 73	PDP-11 20 ~~V96-06807~~ 7527	11-DZRKB-A .BIC (D5HA2)

PROGRAM NAME RK Disk Data Test **DEVICE** RK11

ITEM

Ø, 6/15/73

Phase in - The version of the program is chainable. Under the diagnostic package and incorporated existing MCN's - under chain mode all RK can be tested. The document has been separated from the RK static test.

This MCN is intended to help in formatting a disk-pack. It should be noted that it is very important that the drive and controller be good before an attempt is made to format.

Procedure:

1) Load RK11 disk data test (MAINDEC-11-DZRKB-A-D)
2) Deposit ØØØØØØ in RKDA (777412)
3) Load address = 650 press "start"
4) Put switches 1Ø,1,Ø to one in that order (first 1Ø, then 1, then Ø) rest of the switches to zero.
5) Wait for at least 60 seconds, while the fomatting is being done.
6) Put switches Ø,1,1Ø back to zero in that order only.

The formatting should have been done by now.

NOTE: that in the above procedure there is no checking that the formatting has been done correctly. Hence, the user is advised to check this indirectly using tests ADT1 (SA=6ØØ) or ADT2 (SA=604) in the same diagnostic. Drive Ø has been used in this procedure.

1. 8/14/73

Program clobbers pattern selected when selecting particular pattern to be used in testing. Insert following patch:

LOCATION	FROM	TO	COMMENT
10552	16700	16704	MOV PATNU,R4
10556	10067	10467	MOV R4,ACNVX
10622	22700	22704	CMP #42,R4
10630	16021	16421	MOV PATØ(4), (1)+

MAIN DEC CHANGE NOTICE
MAY BE REQUIRED FOR
PROGRAM TO OPERATE

IDENTIFICATION

PRODUCT CODE	MAINDEC-11-DZRKB-A-D (REPLACES: MAINDEC-11-D5HA-D)
PRODUCT NAME	RK11 DISK DATA TEST
DATE CREATED	MARCH, 1973
MAINTAINER	DIAGNOSTIC GROUP
AUTHOR	E. HAIGHT, C. CASWELL

COPYRIGHT © 1971, 1973
DIGITAL EQUIPMENT CORPORATION

MAINDEC-11-DERKB-A-D
REPLACES D5HA
RK11 DISK DATA

TABLE OF CONTENTS

SECTION	CONTENTS
1,	ABSTRACT
2,	REQUIREMENTS
2,1	EQUIPMENT
2,2	STORAGE
3,	LOADING PROCEDURE
3,1	METHOD
4,	STARTING PROCEDURE
4,1	WORST CASE OPERATION
4,2	OPERATOR INTERVENTION
4,3	SCOPE LOOP ENTRY POINTS
5,	OPERATING PROCEDURE
5,1	CONTROL SWITCH SETTINGS
5,2	CONVERSATION MODE
5,3	SUBROUTINE ABSTRACT
6,	ERROR REPORTS
7,	
7,0	MISCELLANEOUS
7,1	SUGGESTED POWER FAIL TEST

P100

1. ABSTRACT

THE RK11 DISK DATA TEST IS A SERIES OF ADDRESS AND DATA
RELIABILITY ROUTINES WHICH VERIFY TO THE USER THE DISK
CONTROL (RK11) AND DISK (RK02 THRU RK05) ARE OPERATING CORRECTLY.
THIS TEST USED IN CONJUNCTION WITH THE RK11 STATIC TEST
AND RK11 RANDOM DRIVE EXERCISER ASSURES THE USER OF
AN ERROR FREE SYSTEM, WHEN USED IN ITS ENTIRETY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11
RK11 AND (RK02 THRU RK05)

2.2 STORAGE

DATA TEST
MAIN BODY OF PROGRAM OCCUPIES ALL OF MEMORY (MINIMUM REQUIREMENTS 4K WORDS)
THE WRITE AND READ BUFFERS FLOAT THROUGH MEMORY.

3. LOADING PROCEDURE

3.1 METHOD OF LOADING DISK DATA TEST TAPE

PROGRAM FORMAT ABSOLUTE

A. VERIFY THE BOOT LOADER IS IN MEMORY.

B. SET SWITCH REGISTER EQUAL TO #950

MEMORY SIZE	#
4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

C. DEPRESS LOAD ADDRESS.

D. DEPRESS START.

4, STARTING PROCEDURE

4,1 STARTING PROCEDURE COMPLETE DATA TEST

A) SET SWITCH REGISTER EQUAL TO 200
B) DEPRESS LOAD ADDRESS
C) SET SWITCH REGISTER EQUAL TO ZERO
D) DEPRESS START
E) THE PROGRAM WILL THEN REPORT THE NUMBER OF DRIVES ON THE SYSTEM AND WILL CHECK WHETHER THE PROGRAM HAS BEEN LOADED UNDER MONITOR CONTROL BY AN RK DRIVE, IF NOT LOADED BY MONITOR FROM AN RK DRIVE THE TEST WILL START RUNNING BEGINNING WITH DRIVE ZERO

4,2 OPERATOR INTERVENTION FOR DATA TEST ONLY,

A) SET SWITCH REGISTER EQUAL TO 200
B) DEPRESS LOAD ADDRESS
C) SET SWITCH REGISTER EQUAL TO MODE OF OPERATION (REF, SEC5,)
D) DEPRESS START,

4,4,1 ADDRESS ENTRY POINTS FOR TEST ROUTINES

600	ADDRESS TEST 1
604	ADDRESS TEST 2
610	ADDRESS TEST 3
614	ADDRESS TEST 4
620	ADDRESS TEST 5
624	ADDRESS TEST 6
630	RANDOM DRIVE EXERCISER

4,5 ***MAINTENANCE ROUTINES***

634	SELWC	LOAD WORD COUNT REG, WITH SWITCH REGISTER
640	SELCMA	LOAD CURRENT MEMORY ADDR REG, WITH SWITCH REGISTER
644	SELDAR	LOAD DISK ADDR, REGISTER WITH SWITCH REGISTER
650	SELDCS	LOAD DISK CONTROL REGISTER WITH SWITCH REGISTER — — 2003 WRITES 2005 READS
654	TKSEL	LOAD CYLINDER ADDRESS REGISTER FROM SWITCH REGISTER AND SEEK,

4,6 POWER FAIL TESTS

644	PFT1	TEST POWER FAIL DURING WRITE
646	PFT2	TEST POWER FAIL DURING A READ

5, OPERATING PROCEDURE

5,1 CONTROL SWITCH SETTINGS

PROGRAM CONVERSATION

SR15	SET	ENTER PROGRAM CONVERSATION MODE REF, SEC, 5,2
	RESET	OPERATE WORSE CASE TEST ON ALL DRIVES BEGINNING AT DRIVE ZERO

DELETE TYPEOUT

SR14	SET	DELETE TYPEOUTS
	RESET	REPORT MESSAGE

HALT ON READY FLAG

SR13	SET	HALT ON FLAG (READY SET)
	RESET	EXECUTE NEXT OPERATION

(5.1 CONT'D)

DELETE COMPARISIONS

SR12 SET DELETE DATA COMPARISONS

RESET COMPARE DATA BUFFERS

LOOP ON TEST

SR11 SET LOOP ON TEST

RESET CONTINUE TO NEXT TEST

HALT ON ERROR

SR10 SET HALT AFTER ERROR REPORT

RESET CONTINUE AFTER ERROR REPORT

ENTER INTERRUPT BACKGROUND TEST

SR9 SET WAIT FOR INTERRUPTS USING WAIT INSTRUCTION

RESET WAIT FOR INTERRUPTS WITH BACKGROUND TEST

LOOP ON DISK ADDRESS

SR8 SET LOOP ON DISK ADDRESS (SPECIFIED BY PARAMETERS)

RESET CONTINUE TO NEXT DISK BUFFER AREA.

SR7-0 SELECT CYLINDER FROM SR (DURING DYNAMIC TESTING)

CYLINDER SELECTION

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

SELECT ONE OF 312(8) CYLINDERS

5,2 CONVERSATION MODE FOR PROGRAM PARAMETERS FOR DATA TEST ONLY

IN THE PROGRAM CONVERSATION MODE THE OPERATOR CAN SPECIFY ANY ONE OR ALL OF THE PROGRAM PARAMETERS,

PROGRAM CONVERSATION

DATA TEST ONLY?

IF THE OPERATOR ANSWER YES THE PROGRAM WILL ENTER ONLY THE DATA PORTION OF TEST,

MULTI DRIVE MODE?

MULTI DRIVE MODE IS A MODE IN THE PROGRAM WHICH ALLOWS THE OPERATOR TO EXECISE ALL THE DRIVES ON THE SYSTEM WITHOUT RE-STARTING THE PROGRAM, THE PROGRAM AFTER EXERCISING ONE DRIVE WILL REPORT A MESSAGE TELLING THE OPERATOR WHICH DRIVE WILL BE SELECTED NEXT, AND THEN THE PROGRAM WILL EXERCISE THAT DRIVE, WHEN A COMPLETE PASS IS ACCOMPLISHED, A PASS COMPLETE WILL BE REPORTED AND THE TEST WILL RECYCLE,

IF THE OPERATOR ANSWERS "YES" TO THIS QUESTION, HE WILL THEN BE ASKED HOW MANY DRIVES ARE ON THE SYSTEM, AND THEN THE PRECEDING QUESTION WILL BE SKIPPED, IF THE OPERATOR ANSWERS "NO" TO THIS QUESTION, THE NEXT QUESTION WILL BE SKIPPED, AND HE WILL THEN BE ASKED WHICH DRIVE IS TO BE EXERCISED,

NUMBER OF DRIVES 1 TO 10 OCTAL?X

TYPE THE NUMBER OF DRIVES ON THE SYSTEM, FOR MULTI DISK MODE,

EXERCISE DRIVE?X

WHEN NOT IN THE MULTI DRIVE MODE THE OPERATOR WILL HAVE TO SPECIFY WHAT DRIVE IS TO BE USED,

OPTIONAL WORD COUNT(YES-NO)?X

IF THE OPERATOR ANSWERS "NO" TO THIS QUESTION THE NEXT QUESTION WILL BE DELETED FROM THE CONVERSATION,

LENGTH (1 TO SWRDCT)? XXX

THE OPERATOR CAN SPECIFY ANY LENGTH TRANSFER FROM 1(8) TO SWRDCT, THE NORMAL TRANSFER LENGTH VARIES FOR EACH SYSTEM THE NORMAL WORD COUNT (SWRDCT) IS REPORTED EACH TIME THE PROGRAM IS STARTED,

```
        SELECT CYLINDER FROM SR7=Ø?(YES=NO)

IF THE OPERATOR WISHES HE MAY SELECT THE CYLINDER ADDRESS
VIA THE SWITCH REGISTER WHEN RUNNING THE DATA TEST BY
ANWSERING YES TO THIS QUESTION,

        STARTING SURFACE AND SECTOR? X

IF THE OPERATOR WISHES TO START A TRANSFER AT A PARTICULAR
SURFACE AND SECTOR HE MAY BY SPECIFYING THOES PARAMETERS HERE,
        *THE MAX, VALUE FOR THE SURFACE AND SECTOR MAY NOT
         EXCEED 33(8), THE SECTORS ON SURFACE ZERO RUN
         FROM Ø TO 13(8), THE SECTORS ON SURFACE ONE RUN FROM
         2Ø(8) TO 33(8),
```

(5.2 CONT'D)

OPTIONAL DATA PATTERN NO.?X

IF NO OPTIONAL DATA PATTERN IS REQUESTED THE PROGRAM WILL EXECUTE THE FOLLOWING LIST OF DATA PATTERNS,

PATTERN	0	= 000000
"	1	= 177777
"	2	= 134510
"	3	= 043267
"	4	= 100000
"	5	= 107070
"	6	= 070707
"	7	= 052525
"	10	= 125252
"	11	= 177737
"	12	= 004102
"	13	= 136363
"	14	= 063636
"	15	= 000001
"	16	= 100005
"	17	= 000520
"	20	= 030303
"	21	= RANDOM DATA
"	22	= RUN ALL DATA PATTERNS UNDER PROGRAM CONTROL

IF THE OPERATOR DECIDES TO OPERATE UNDER PROGRAM CONTROL, THE NEXT QUESTION WILL BE SKIPPED. THE OPERATOR BY REQUESTING AN OPTIONAL DATA PATTERN HAS THE ABILITY OF SELECTING ANY ONE OF THE 21(8) DATA PATTERNS. DATA PATTERN NO, XX

CONTROL PROGRAM FUNCTION PARAMETERS

IN THIS SECTION OF THE PROGRAM PARAMETER CONVERSATION MODE, THE OPERATOR CAN SELECT ANYONE OR ALL THREE OF THE CONTROL FUNCTIONS TO BE EXECUTED. THE NORMAL SEQUENCE OF CONTROL FUNCTIONS UNDER PROGRAM CONTROL ARE WRITE, READ, READ CHECK AND WRITE CHECK. BY ENTERING THE CONVERSATION MODE THE OPERATOR HAS GAINED COMPLETE CONTROL OVER THE CONTROL FUNCTIONS. HE MUST SPECIFY YES OR NO TO ALL OF THE FOLLOWING QUESTIONS,

WRITE?	(YES = NO)
READ?	(YES = NO)
READ CHECK?	(YES = NO)
WRITE CHECK?	(YES = NO)

TO PERFORM A WRITE CHECK ONLY, THE OPERATOR MUST FIRST WRITE SOME KNOWN DATA ON THE DISK. THIS COURSE OF ACTION ALSO PREVAILS FOR A READ ONLY OPERATION,

* IF AN ERROR OCCURS IN THE LINE THE OPERATOR IS TYPING, DEPRESS THE RUB-OUT. THIS CAUSES THE QUESTION TO BE RETYPED AND ALLOWS THE OPERATOR TO PROPERLY ANSWER THE QUESTION,

* INDICATES TO THE OPERATOR THAT THE CARRIAGE-RETURN SHOULD BE TYPED AT THE INDICATED PLACE TO TERMINATE THE LINE OF TYPED CHARACTERS,

5,3, SUBROUTINE ABSTRACTS

ADDRESS TEST 1 (ADT1)

IN THIS TEST THE PROGRAM WRITES EACH SECTOR ON
ITSELF AND THEM READS IT BACK AND VERIFIES THAT THE
CORRECT SECTOR WAS READ, THIS TEST USES A BLOCK LENGTH
OF 400 WORDS,

ADDRESS TEST 2 (ADT2)

IN THIS TEST THE ABILITY OF THE CONTROL AND DRIVE
TO SEEK FROM CYLINDER 0 TO EACH OTHER CYLINDER
IS CHECKED, ONLY SEEKS ARE USED NOT RESTORES,

ADDRESS TEST 3 (ADT2UX)

THIS THIS TEST THE CYLINDER SELECTION MATRIX IS
CHECKED BY WRITING THE FIRST WORD OF EACH CYLINDER
WITH ITS OWN CODE,THEN THE PROGRAMS STARTS SEEKING FROM
CYLINDER 312 TO ALL OTHER CYLINDERS TO VERIFY
CORRECT SELECTION,

ADDRESS TEST 4 (ADT3UX)

IN THIS TEST THE ABILITY OF THE DRIVE TO DO MAX,
HEAD MOVEMENT IS CHECKED, IN THIS THE DRIVE FIRST
SEEKS TO 0 THEN TO CYLINDER 312, THEN BOTH THE
COUNTS ARE ALTERED INSTEAD OF SEEKING TO 0
THIS TIME WE SEEK TO 1,AND INSTEAD SEEKING TO
312 WE SEEK TO 311, THIS CONTINUES UNTIL 0 INCREMENTS
TO 312,AND THE COUNT THAT START AT 312 GOES TO 0,

ADDRESS TEST 5 (SAWTH)

IN THIS TEST THE PROGRAM CREATES HEAD MOVEMENT WHICH
APPEARS AS A SAW TOOTH PATTERN, THIS IS TO CREAT
A WORST CASE NOISE,

ADDRESS TEST 6 (LOOK AHEAD)

IN THIS TEST THE PROGRAM CHECKS THE ABILITY OF THE DRIVE
TO LOCATE THE SECTOR ADDRESS ,

RANDOM DRIVE EXERCISER (RANEX)

IN THIS TEST THE ABILITY OF THE DRIVE TO DO
RANDOM SEEKS AND TO WRITE AND READ RANDOM DATA AT THE
SAME TIME IS CHECKED, THE WORD COUNT FOR THIS TEST
IS 400 WORDS OR ONE BLOCK.

MAINTENANCE TOOL

TKSEL = STATIC CYLINDER SELECTION

THIS ROUTINE WAS DESIGNED TO ENABLE THE OPERATOR TO HAVE A QUICK
METHOD OF SELECTING CYLINDER AND ISSUING A SEEK.

(5.3 CONT'D)

TKSEL = OPERATING PROCEDURE

STEP A. SET SWITCH REGISTER EQUAL TO 654

STEP B. DEPRESS LOAD ADDRESS

STEP C. SET SWITCH REGISTER 10 THRU 8 EQUAL TO DRIVE #0 THRU 7

STEP D. DEPRESS START

STEP E. SET SWITCH REGISTER 7 THRU 0 EQUAL TO CYLINDER #

STEP F. DEPRESS CONTINUE

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

CYLINDER NUMBER
(0 THRU 177(8))

STEP G. TRACK NUMBER CAN BE CHANGED ARBITRARILY.

6. ERROR REPORTS

6.1 ADDRESS ERROR

XX ERR CNT XXXXXXWRD1XXXXWRD2

ERRCNT = IS THE TAG FOR THE LISTING
WRD1 = WHAT WAS EXPECTED
WRD2 = WHAT WAS RECEIVED

WHEN A REPORT ONLY CONTAINS ONE WORD THE PROGRAM WAS EXPECTING
ZEROS BUT RECEIVED WHAT WAS REPORTED.

6.2 ERROR REPORTS

STATUS ERROR

STATUS ERROR XXXXXXRKDA XXXXXXRKER XXXXXXRKCS

A B C

A=THE DRIVE ADDRESS REGISTER
B=THE CONTROL ERROR REGISTER
C=THE CONTROL FUNCTION REGISTER

BIT LAYOUT OF RKCS REGISTER
BIT15= ERROR
BIT14= HARD ERROR
BIT13= SEARCH COMPLETE
BIT12= MAINTENANCE ENABLE
BIT11= NOT USED
BIT10= FORMAT
BIT9 = READ/WRITE ALL
BIT8 = STOP ON SOFT ERROR
BIT7= READY
BIT6= INTERRUPT ENABLE
BIT5= EXTENDED MEMORY 1 (XM1)
BIT4= EXTENDED MEMORY 0 (XM0)
BIT3-2-1= FUNCTION REGISTER
BIT0= GO

BIT 3	BIT 2	BIT 1	OPERATION
0	0	0	RESET
0	1	0	READ
0	0	1	WRITE
0	1	1	WRITE CHECK
1	0	0	SEEK
1	0	1	READ CHECK
1	1	0	DRIVE RESET
1	1	1	WRITE LOCK

BIT LAYOUT OF RKER REGISTER

BIT15= DRIVE ERROR (DRE)
BIT14= OVERRUN (OVR)
BIT13= WRITE LOCK OUT VIOLATION (WLO)
BIT12= SEEK ERROR (SKE)
BIT11= PROGRAMMING ERROR (PCE)
BIT10= NON-EXISTENT MEMORY ERROR (NXM)
BIT9 = DATA LATE ERROR (DTL)
BIT8 = READ TIMING ERROR (RTE)
BIT7 = NON-EXISTENT DISK ERROR (NXD)
BIT6 = NON-EXISTENT CYLINDER ERROR (NXC)
BIT5 = NON-EXISTENT SECTOR (NXS)
BIT4-3-2= NOT USED
BIT1 = CHECKSUM ERROR (CSE)
BIT0 = WRITE CHECK ERROR (WCE)

IF THE ERROR OCCURRED WHEN READING THE PROGRAM WILL REPORT
WHICH READ,

NOTE! WHEN A HARD ERROR OCCURS AN ADDITIONAL ERROR MESSAGE
WILL BE REPORTED, AS FOLLOWS:

XXXXXX RKDS

LAYOUT OF RKDS BITS

BIT15-14-13= IDENTIFICATION OF DRIVE
BIT12= DRIVE POWER LOW (DPL)
BIT11= HIGH DENSITY DRIVE (HDEN)
BIT10= DRIVE UNSAFE (DRU)
BIT9 = SEEK INCOMPLETE (SIN)
BIT8 = SECTOR COUNTER OK (SOK)
BIT7 = DRIVE READY (DRY)
BIT6 = ACCESS READY (ARDY)
BIT5 = WRITE PROTECT STATUS (WPS)
BIT4 = SECTOR COUNTER EQUALS SECTOR ADDRESS (SC=SA)
BIT3-0= SECTOR COUNTER (C_3)

6,3 DATA ERRORS

1 READ XXXXXXRKDA XXXXXXGD DATA XXXXXXBD DATA

A B C D

A=WHICH READ THE ERROR OCCURRED
B=THE DRIVE ADDRESS REGISTER
C=THE DATA WRITTEN ON THE DRIVE SURFACE
D=THE DATA READ FROM THE DRIVE SURFACE

6,4 RANEX ERRORS

ERRORS WHICH OCCUR IN THE RANDOM EXERCISER (RANEX) WILL ALSO HAVE THE FOLLOWING MESSAGE REPORTED.

RANEX ERROR

6,7 PROCESSOR TIME OUT

CPU BKGRND TIMED OUT

THIS MESSAGE WILL BE REPORTED IF THE RK11 CONTROL FAILS TO RAISE A BR REQUEST AFTER EXTENDED PERIOD OF TIME.

6,8,1 MULTIPLE DRIVE OPERATION

NUMBER OF DRIVES Xx

THIS MESSAGE IS REPORTED AT THE START OF THE PROGRAM TO INDICATE THE TOTAL NUMBER OF SYSTEM DRIVES.

NOTE! THE DRIVES MUST BE IN NUMERICAL SEQUENCE.

6,8,2 STANDARD WORD COUNT REPORT (SWRDCT)

XXXXXXSWRDCT

THIS MESSAGE IS REPORTED AT THE START OF THE PROGRAM TO TELL THE OPERATOR WHAT THE LARGEST TRANSFER IS THAT HE MAY MAKE TO THE RK11 DRIVE, THIS WORD COUNT VARIES FROM SYSTEM TO SYSTEM SINCE THE BUFFERS FLOAT THROUGH MEMORY.

6,8 END

END

THIS MESSAGE IS REPORTED AT THE END OF ONE COMPLETE PASS OF THE DISK SYSTEM.

6,9 PARITY ERROR

THIS MESSAGE IS REPORTED IF THE PROGRAM DETECTS A PARITY ERROR DURING PROGRAM EXECUTION.

7. MISCELLANEOUS

IN SOME ADDRESS TESTS THE PROGRAM DEPENS ON WRITTING AND READING DATA CORRECTLY FROM THE DISK,AND IF IT DOES NOT IT MAY REPORT AN ADDRESS FAILURE, WHEN IN FACT IT WAS A DATA FAILURE.

DZRKBA,BIC

```
1
2                       ;TITLE  MAINDEC-11-DZRKB-A      RK11 DATA TEST  REPLACES D5HA-A
3                       ;COPYRIGHT 1973, DIGITAL EQUIPMENT  CORP.,  MAYNARD, MASS.
4                       ;PROGRAM BY E. HAIGHT/C. CASWELL
```

```
                                                        ;ABS
                    000001          BI=0=1
                    000002          BI=1=2
                    000004          BI=2=4
                    000010          BI=3=10
                    000020          BI=4=20
                    000040          BI=5=40
                    000100          BI=6=100
                    000200          BI=7=200
                    000400          BI=8=400
                    001000          BI=9=1000
                    002000          BI=10=2000
                    004000          BI=11=4000
                    010000          BI=12=10000
                    020000          BI=13=20000
                    040000          BI=14=40000
                    100000          BI=15=100000
                                    ;
                                    ;
                                    ;
                    104403          WRITE=TRAP+3
                    104407          WRCHECK=TRAP+7
                    104405          READ=TRAP+5
                    104413          RDCK=TRAP+13
                    000240          NOP=240
                    000000          .=0
                                    .REPT   100
                                    .+2
                                    HALT
                                    .ENDR
                    000020                  .=20
000020  011714                      XWAIT
000022  000000                      0
000024  012142                      DOWN
000026  000340                      340
                    000030          .=30
000030  012502                      EMTRP
000032  000340                      340
000034  006646                      DISK
000036  000340                      340
                    000042          .=42
000042  000000                      0
                    000200          .=200
000200  000167  000714      JMP     START
```

```
                                                                    ;
                                                                    ;
                    000600                      .=600
000600  000167  001474                  JMP     ADT1                ;WRITE EACH SECTOR ON ITSELF AND
                                                                    ;READ IT BACK TO VERIFY IT
                                                                    ;                          ;
000604  000167  002120                  JMP     ADT2                ;TEST THE ABILITY ON THE DRIVE TO SEEK
                                                                    ;FROM CYLINDER TO ALL OTHER CYLINDERS
                                                                    ;
                                                                    ;
000610  000167  002550                  JMP     ADT2UX              ;TEST THE CYLINDER SELECTION MATRIX
                                                                    ;
                                                                    ;
000614  000167  003250                  JMP     ADT3UX              ;THIS TEST IS A MAX, HEAD MOVEMENT TEST;
                                                                    ;THE TEST SEEKS FROM 0 TO 312 THEN
                                                                    ;FROM 312 TO 0 THEN IT DEC', THE HIGH
                                                                    ;COUNT AND INC', THE LOW COUNT AND
                                                                    ;SEEKS TIL 0 BECOMES 312 AND HIGH BECOMES 0
                                                                    ;
                                                                    ;
000620  000167  003646                  JMP     SAWTH               ;IN THIS TEST THE HEADS APPEAR TO
                                                                    ;BE MOVING IN A SAW TOOTH PATTERN
                                                                    ;WHICH IS TO GENERATE WORST CASE NOISE
                                                                    ;
                                                                    ;
000624  000167  004314                  JMP     ADT3                ;THIS TEST CHECKS THE LOOK AHEAD FEATURE
                                                                    ;OF THE CONTROL
                                                                    ;
                                                                    ;
000630  000167  005250                  JMP     RANEX               ;RANDOM ADDRESS, DATA AND
                                                                    ;WORD COUNT TEST
                                                                    ;
                                                                    ;
                    000634                                          .=634

                                ;***MAINTENANCE ROUTINES***
                                ;
000634  000167  011202                  JMP     SELWC               ;LOAD WORD COUNT REG,
                                                                    ;WITH SWITCH REGISTER
                                                                    ;
                                                                    ;
000640  000167  011206                  JMP     SELCMA              ;LOAD CURRENT MEMORY ADDR
                                                                    ;REG, WITH SWITCH REGISTER
                                                                    ;
                                                                    ;
000644  000167  011212                  JMP     SELDAR              ;LOAD DISK ADDR, REGISTER
                                                                    ;WITH SWITCH REGISTER
                                                                    ;
                                                                    ;
000650  000167  011216                  JMP     SELDCS              ;LOAD DISK CONTROL REGISTER
                                                                    ;WITH SWITCH REGISTER
                                                                    ;
                                                                    ;
000654  000167  011332                  JMP     TKSEL               ;ENABLE READ AMPLIFIERS
```

```
                                                        ;TO TRACK SELECTED
                                                        ;FROM SWITCH REGISTER
                                                        ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;RK11 DATA TEST
                                ;VECTORS USED IN PROGRAM
                                ;#1 LOC 220 DISK INTERRUPT
                                ;#2 LOC 30 EMT (TELETYPE OUTPUT)
                                ;#3 LOC 34 TRAP (DISK HANDLERS)
                                ;#4 LOC 14 TRACE TRAP (USED IN BACKGROUND TEST)
                                ;#5 LOC 20 IOT TRAP (USED IN CALLING BACKGROUND TEST)

001000                          .=1000
```

```
                                        .EVEN
                                ;
                                ;I/O ADDRESS POINTERS
001000  177570                  SWR:    177570                  ;SWITCH REGIISTER
001002  177776                  PS:     177776                  ;PROCESSOR STATUS REGISTER
001004  177566                  TPB:    177566                  ;TELETYPE REGISTERS
001006  177562                  TKB:    177562
001010  177564                  TPS:    177564
001012  177560                  TKS:    177560
                                ;
                                ;DISK I/O REGISTERS
                                ;
001014  177404                  RKCS:   177404                  ;DISK CONTROL REGISTER
001016  177406                  RKWC:   177406                  ;WORD COUNT REGISTER
001020  177410                  RKBA:   177410                  ;CURRENT ADDRESS REGISTER
001022  177412                  RKDA:   177412                   ;DISK ADDRESS REGISTER
001024  177402                  RKER:   177402                  ;ERROR REGISTER
001026  177400                  RKDS:   177400                  ;DRIVE STATUS REGISTER
001030  177416                  RKDB:   177416                  ;DATA BUFFER REGISTER
001032  177414                  RKMR:   177414                  ;MAINTENANCE REGISTER
001034  000220                  VECTOR: 220                     ;INTERRUPT VECTOR ADDRESS
001036  000222                  STATUS: 222                     ;DISK INTERRUPT STATUS
001040  000200                  PRIORITY:BIT7                   ;DISK PRIORITY LEVEL

                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;RK11 DEDICATE REGISTERS (MEMORY)
                                ;
001042  000000                  FLAG:   0                       ;INTERNAL PROGRAM FLAG
001044  146723                  RANNU:  146723                  ;RANDOM NUMBER PRIME
001046  000000                  WRDCT:  0                       ;WORKING WORD COUNT
001050  000000                  TRACK:  0                       ;WORKING DAE
001052  000000                  DMA:    0                       ;WORKING DAR
001054  000000                  PATNU:  0                       ;DATA PATTERN INDEX
001056  000000                  BUF:    0                       ;WORKING DATA BUFFER (OUT-IN)
001060  000000                  TWRDCT: 0                       ;TEMP WORD COUNT
001062  000000                  TDMA:   0                       ;TEMP DAR
001064  000000                  SWRDCT: 0                       ;STANDARD WORD COUNT
001066  000000                  ERCOUNT:0                       ;ERROR COUNT FOR MESSAGES,
001070  000000                  SAVE:   0
001072  000000                  SAV11:  0
001074  000000                  PASS:   0
001076  000000                  DSKNOR: 0
001100  000000                  HRDER:  0                       ;POINTER FOR HARD ERROR
001102  000000                  BLOCK:  0
001104  000000                  PASSC:  0
001106  000000                  INBUF:  0

                                ;
```

```
                                        ;RK11 WORK REGISTERS
                                        ;(CAN BE CHANGED IN ANY ROUTINE)
001110  000000                          WORK:   0
001112  000000                          WORK1:  0
001114  000000                          WORK2:  0
001116  000000                          WORK3:  0
```

```
                                                                
001120  000005                  START:  RESET                   ;CLEAR THE WORLD
001122  012767  000010  177760          MOV     #10,WORK
001130  012700  013450                  MOV     #TABLE,R0       ;CLEAR ERROR TABLE
001134  005020                  CLRTAB: CLR     (R0)+
001136  005367  177746                  DEC     WORK
001142  001374                          BNE     CLRTAB
001144  012706  001000                  MOV     #1000,R6        ;SET UP STACK
001150  004567  010276                  JSR     R5,EXTMEN       ;SET UP DATA BUFFERS
001154  012706  001000                  MOV     #1000,R6        ;RESET STACK
001160  012777  000340  177614          MOV     #340,@PS        ;LOCK UP INTERRUPT LEVELS
001166  005067  177650                  CLR     FLAG            ;CLEAR PROGRAM FLAG
001172  005067  177654                  CLR     DMA             ;CLEAR DAR REGISTERS
001176  005067  177652                  CLR     PATNU           ;CLEAR PATTER COUNT
001202  016767  177656  177636          MOV     SWRDCT,WRDCT
001210  004767  012032                  JSR     R7,MAMF         ;SET PARITY SWITCHES
001214  005777  177560                  TST     @SWR
001220  100463                          BMI     LCONM           ;OPERATE UNDER PROGRAM CONTROL
001222  052767  070100  177612          BIS     #70100,FLAG
001230  005067  012010                  CLR     TEXBUF
001234  105777  177566          SPECOR: TSTB    @RKDS
001240  100013                          BPL     RKNODV
001242  022767  000010  011774          CMP     #10,TEXBUF
001250  001407                          BEQ     RKNODV
001252  062767  000001  011764          ADD     #1,TEXBUF
001260  062777  020000  177534          ADD     #20000,@RKDA
001266  000762                          BR      SPECOR
                                                                
001270  005767  011750          RKNODV: TST     TEXBUF
001274  001003                          BNE     REPDRV
001276  104001                          EMT     +1
001300  014674                          NODRV
001302  000000                          HALT
001304  016767  011734  177564  REPDRV: MOV     TEXBUF, DSKNOR  ;STORE # OF DISKS
001312  162767  000001  177556          SUB     #1, DSKNOR      ;SUB 1 FOR FAILED TST
001320  006167  177552                  ROL     DSKNOR          ;SHIFT LEFT TO
001324  006167  177546                  ROL     DSKNOR          ;MATCH POSITION IN FLAG
001330  004567  011416                  JSR     R5, CONV
001334  013244                          TEXBUF
001336  014722                          RKNUM
001340  000002                          2
001342  104000                          EMT     +0
001344  013552                          HED5A
001346  014722                          RKNUM
001350  177777                          -1
001352  052767  004000  177462          BIS     #BIT11,FLAG     ;SET UP PROGRAM FLAG
001360  005077  177436                  CLR     @RKDA           ;RESET DISK ADDRESS
001364  000167  000002                  JMP     AUTST
                                ;ENTER OPERATOR CONVERSATION MODE
001370  004567  011656          LCONM: JSR      R5,     CONV    ;CONVERT SWRDCT TO ASCII
001374  001064                          SWRDCT
001376  014125                          SPECMES
001400  000006                          6
001402  104000                          EMT+0
```

```
                                        HED$A
                                        SPECMES
                                        *1
CONM1:  EMT     +1                      ;ASK ABOUT DATA TEST ONLY
                CON1
        JSR     %7,ALPHA                ;GO WAIT FOR ANSWER
        CMP     #153,TEXBUF             ;TEST FOR YES
        BNE     .+10                    ;BRANCH IF NO
        BIS     #BIT10,FLAG
        EMT     +1
                CON2                    ;ASK ABOUT MULTI DRIVE MODE
        JSR     %7,ALPHA                ;GO WAIT FOR ANSWER
        CMP     #153,TEXBUF             ;TEST FOR YES
        BNE     DATTES                  ;
        BIS     #BIT11,FLAG             ;
DSKDR1: EMT     +1
                CON3
        JSR     %7,NOCHA
        SUB     #1,TEXBUF
        CMP     #10,TEXBUF
        BLOS    DSKDR
        MOV     TEXBUF,DSKNOR
        ROL     DSKNOR
        ROL     DSKNOR
        BR      ASKWC
DATTES: EMT     +1
                CON4                    ;ASK UNIT NUMBER
        JSR     %7,NOCHA                ;WAIT FOR NO,
        CMP     #10,TEXBUF              ;IS NO = OR>10
        BLOS    DATTES                  ;NO
        CLC
        ROL     TEXBUF
        ROL     TEXBUF
        BIS     TEXBUF,FLAG
ASKWC1: EMT     +1
                CON5                    ;ASK ABOUT OPTIONAL WORD COUNT
        JSR     %7,ALPHA                ;WAIT FOR ANSWER
        CMP     #153,TEXBUF
        BNE     TKSR                    ;ASK ABOUT OPTIONAL DAR
WCCON1: EMT     +1
                CON6                    ;ASK LENGTH OF WC
        JSR     %7,NOCHA
        TST     TEXBUF
        BEQ     WCCON
        MOV     SWRDCT,WORK
        INC     WORK
        CMP     WORK,TEXBUF             ;IS TEXBUF LESSTHAN SWRDCT
        BLOS    WCCON                   ;YES ASK FOR COUNT AGAIN
        MOV     TEXBUF,SWRDCT           ;OPERATING WORD COUNT
        MOV     SWRDCT,WRDCT
TKSR1:  EMT+1                           ;ASK ABOUT CYCINDER #
                CON7A
        JSR     %7,ALPHA
        CMP     #153,TEXBUF
```

```
   293  001706  001003                        BNE     OPDAR
   294  001710  052767  000040  177124        BIS     #BIT5,FLAG
   295  001716  104001                OPDAR1  EMT     +1
   296  001720  014377                        CON7                    ;ASK ABOUT OPTIONAL DAR
   297  001722  004767  011106                JSR     %7,NOCHA
   298  001726  022767  000034  011310        CMP     #34,TEXBUF      ;33 MAX SECTOR COUNT
   299  001734  101770                        BLOS    OPDAR
   300  001736  016767  011302  177106        MOV     TEXBUF, DMA     ;TEMP SECTOR REGISTER
   301  001744  042767  177740  177100        BIC     #177740,DMA
```

```
                302
000752  104001  303                 OPPAT:  EMT     +1
000754  014503  304                         CON8                    ;ASK ABOUT DATA PATTERNS
000756  004767  305         011052          JSR     %7,NOCHA
000762  022767  306         000023  011254  CMP     #23,TEXBUF      ;TEST FOR CORRECT NO
000770  101770  307                 BLOS    OPPAT                   ;ASK AGAIN
000772  022767  308         000022  011244  CMP     #22,TEXBUF
002000  001414  309                 BEQ     OPWRT                   ;DATA PATTERN UNDER PROGRAM CONTROL
002002  052767  310         100000  177032  BIS     #BIT15,FLAG     ;SET PROGRAM FLAG
002010  016767  311         011230  177036  MOV     TEXBUF, PATNU   ;OPERATOR WANTS TO SELECT DATA
002016  000241  312                 CLC
002020  006167  313         177030          ROL     PATNU
002024  042767  314         070100  177010  BIC     #70100,FLAG     ;CLEAR OP MODE BITS IN FLAG
002032  104001  315         OPWRT:  EMT     +1
002034  014537  316                 CON9                    ;ASK ABOUT WRITE
002036  004767  317         011002          JSR     %7,ALPHA
002042  022767  318         000153  011174  CMP     #153,TEXBUF     ;TEST FOR YES
002050  001003  319                 BNE     OPRD                    ;ASK ABOUT READ
002052  052767  320         040000  176762  BIS     #BIT14,FLAG     ;YES SET FLAG BIT
002060  104001  321         OPRD:   EMT     +1
002062  014605  322                 CON11                   ;ASK ABOUT READ
002064  004767  323         010754          JSR     %7,ALPHA
002070  022767  324         000153  011146  CMP     #153,TEXBUF     ;TEST FOR YES ANSWER
002076  001003  325                 BNE     RDCHK
002100  052767  326         010000  176734  BIS     #BIT12,FLAG     ;SET FLAG TO READ
002106  104001  327         RDCHK:  EMT+1
002110  014624  328                 CON12
002112  004767  329         010726          JSR     %7,ALPHA
002116  022767  330         000153  011120  CMP     #153,TEXBUF
002124  001003  331                 BNE     OPWCK
002126  052767  332         000100  176706  BIS     #BIT6,FLAG
002134  104001  333         OPWCK:  EMT+1
002136  014557  334                 CON10                   ;ASK ABOUT WRITE CHECK
002140  004767  335         010700          JSR     %7,ALPHA
002144  022767  336         000153  011072  CMP     #153,TEXBUF
002152  001003  337                 BNE     .+10
002154  052767  338         020000  176660  BIS     #BIT13,FLAG
002162  032767  339         070100  176652  BIT     #70100,FLAG
002170  001720  340                 BEQ     OPWRT
                341                 ;
                342                 ;
                343                 ;
002172  032767  344         004000  176642  ADTST:  BIT     #BIT11,FLAG     ;ARE WE IN MULTI DRIVE MODE
002200  001431  345                 BEQ     EXMFLG                  ;BRANCH IF NO,
002202  122767  346         000002  175631          CMPB    #2,41           ;IS THIS AN RK11 MONITOR LOAD?
002210  001002  347                 BNE     1$                      ;NO - CONTINUE
002212  000167  348         004312          JMP     CHKDOS                  ;GO TO NEXT DISK
002216  104001  349         1$:     EMT     +1
002220  014003  350                 MES11
002222  016767  351         176614  010602  MOV     FLAG,ACNVX
002230  006067  352         010576          ROR     ACNVX
002234  006067  353         010572          ROR     ACNVX
002240  042767  354         177770  010564  BIC     #177770,ACNVX   ;FETCH DRIVE #
002246  004567  355         010500          JSR     %5,CONV
```

```
  356  002252  013032                          ACNVX
  357  002254  014015                          MES12
  358  002256  000001                          1
  359  002260  104001                          EMT     +1
  360  002262  014015                          MES12
  361  002264  032767  002000  176550  EXMFLG: BIT     #BIT10,FLAG     ;TEST FOR DATA TEST ONLY
  362  002272  001402                          BEQ     .+6             ;DO COMPLETE TEST
  363  002274  000167  003110                  JMP     DATAT           ;DO DATA TEST ONLY
```

```
                                        ;
                                        ;***** ADDRESS TEST *****
                                        ;
                                        ;
                                        ;RK11 ADDRESS TEST #1 (TRACK AND SECTOR SELECTION TEST)
                                        ;
                                        ;WRITE 200(OCTAL) WORDS IN EACH SECTOR
                                        ;THE WORD CONTAINS THE ADDRESS OF EACH SECTOR
                                        ;WHEN THE COMPLETE DISK IS WRITTEN READ
                                        ;BACK EACH SECTOR AND COMPARE FOR THE CORRECT
                                        ;DATA IN THE SECTOR
002300  016767  176560  176562  ADT1:  MOV     $WRDCT,SAVE
002306  004567  006022                  JSR     R5,DSKNOS       ;SET UP DISK NUMBER
002312  056777  176572  176502          BIS     WORK,@RKDA      ;SET UP DRIVE NUMBER
002320  012777  000015  176466          MOV     #15,@RKCS       ;RESET DRIVE
002326  105777  176462                  TSTB    @RKCS           ;WAIT FOR DRIVE READY
002332  100375                          BPL     .-4
002334  032777  000100  176464          BIT     #BIT6,@RKDS
002342  001774                          BEQ     .-6
002344  012767  000200  176512          MOV     #200,$WRDCT
002352  012767  000200  176466          MOV     #200,WRDCT      ;SETUP WORD COUNT
002360  012767  014750  176470          MOV     #OUTBUF,BUF     ;SET UP CURRENT ADDRESS
002366  012706  001000                  MOV     #1000,R6        ;SET UP STACK
002372  005067  176454                  CLR     DMA
002376  012777  000001  176410  SEABUF: MOV     #BIT0,@RKCS
002404  105777  176404                  TSTB    @RKCS
002410  100375                          BPL     .-4
002412  012700  014750                  MOV     #OUTBUF,R0      ;SET UP ADDRESS BUFFER
002416  016701  176424                  MOV     WRDCT,R1        ;
002422  016720  176424          XSEABUF: MOV    DMA,(0)+
002426  005301                          DEC     R1
002430  001374                          BNE     XSEABUF
002432  104403                          WRITE                   ;WRITE SECTOR
002434  105777  176354                  TSTB    @RKCS           ;CHECK FOR READY
002440  100375                          BPL     .-4
002442  005777  176346                  TST     @RKCS           ;TEST FOR ERROR
002446  100014                          BPL     WRNEXB          ;BRANCH IF NO ERROR
002450  012767  000001  176410          MOV     #1,ERCOUNT
002456  017767  176332  176424          MOV     @RKCS,WORK      ;FETCH CONTENTS OF CONTROL REG
002464  017767  176336  176420          MOV     @RKDS,WORK1     ;FETCH CONTENTS OF ADDRESS REG
002472  004567  007724          ERR1:   JSR     R5,STAER1       ;REPORT ERROR
002476  000737                          BR      SEABUF          ;LOOP ON ERROR
002500  004767  005120          WRNEXB: JSR     R7,DISBUF       ;SET UP NEXT DISK ADDR;
002504  000734                          BR      SEABUF          ;WRITE NEXT SECTOR
002506  005067  176340                  CLR     DMA
002512  016767  176370  176336  SETADT: MOV     INBUF,BUF       ;SET UP CURRENT ADDRESS
002520  012777  000015  176266          MOV     #15,@RKCS       ;RESET DRIVE
002526  105777  176262                  TSTB    @RKCS
002532  100375                          BPL     .-4
002534  032777  000100  176264          BIT     #BIT6,@RKDS
002542  001774                          BEQ     .-6
```

```
                                                                    
002544  012777  000001  176242  RDSECT: MOV     #BIT0,@RKCS
002552  105777  176236                  TSTB    @RKCS
002556  100375                          BPL     .-4
002560  104405                          READ
002562  105777  176226                  TSTB    @RKCS           ;CHECK FOR READY
002566  100375                          BPL     .-4             ;NOT READY BRANCH BACK
002570  005777  176220                  TST     @RKCS           ;TEST FOR ERROR
002574  100014                          BPL     ADWGT           ;BRANCH IF NO ERROR
002576  017767  176212  176304          MOV     @RKCS,WORK      ;FETCH CONTROL REG
002604  017767  176212  176300          MOV     @RKDA,WORK1     ;FETCH DISK ADDR
002612  012767  000002  176246          MOV     #2,ERCOUNT      ;SET UP ERROR COUNT
002620  004567  007576          ERR2:   JSR     %5,STAER1       ;REPORT ERROR
002624  000747                          BR      RDSECT          ;LOOP ON ERROR
002626  016700  176254          ADWGT:  MOV     INBUF,%0
002632  012701  000200                  MOV     #200,%1
002636  026710  176210          SANHT:  CMP     DMA,(0)         ;CMP FOR CORRECT ADDR;
002642  001004                          BNE     ADERR           ;BRANCH IF DATA DID NOT COMPARE
002644  005720                  INCADT: TST     (0)+
002646  005301                          DEC     %1
002650  001372                          BNE     SANHT           ;TEST NEXT WORD
002652  000412                          BR      CHKADT
002654  016767  176172  176230  ADERR:  MOV     DMA,WORK1       ;CORRECT ADDRESS
002662  011067  176222                  MOV     (0),WORK        ;DATA IN ERROR
002666  012767  000003  176172          MOV     #3,ERCOUNT      ;SET UP ERROR COUNT
002674  004567  007522          ERR3:   JSR     %5,STAER1       ;REPORT ERROR
                                ;*****REPORT ONLY ONE ERROR PER SECTOR*****
002700  004767  004720          CHKADT: JSR     %7,DISBUF       ;SET UP NEXT DISK BUFFER
002704  000717                          BR      RDSECT          ;CHECK NEXT SECTOR
002706  016767  176156  176150          MOV     SAVE,SWRDCT     ;RESTORE STANDARD WORD COUNT
002714  032777  004000  176056          BIT     #BIT11,@SWR     ;DOES OPERATOR WANT TO LOOP ON TEST
002722  001402                          BEQ     .+6
002724  000167  177350                  JMP     ADT1
```

```
                                        ;ADDRESS TEST #2
                                        ;POLE TEST
                                        ;
                                        ;SEEK FROM CYLINDER 0 TO ALL OTHER CYLINDERS
                                        ;
002730  005067  176116          ADR21   CLR     DMA
002734  012777  000001  176052          MOV     #BIT0,@RKCS     ;RESET CONTROL
002742  105777  176046          TSTB    @RKCS
002746  100375                  BPL     .-4
002750  012706  001000          MOV     #1000,%6        ;SET UP STACK
002754  012777  005116  176052  MOV     #PWAIT,@VECTOR  ;SET UP INTERRUPT VECTOR
002762  012777  000340  176046  MOV     #340,@STATUS
002770  016777  176044  176004  MOV     PRIORITY,@PS
002776  005077  176020          CLR     @RKDA
003002  004567  005326          JSR     %5,DSKNOS       ;SET UP THE DISK
003006  056777  176076  176006  BIS     WORK,@RKDA      ;SET UP THE DRIVE NUMBER
003014  012777  000011  175772  MOV     #11,@RKCS       ;SEEK FOR ZERO
003022  105777  175766          TSTB    @RKCS           ;TEST FOR CONTROL READY
003026  100375                  BPL     .-4
003030  032777  000100  175770  BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003036  001774                  BEQ     .-6
003040  056777  176006  175754  BIS     DMA,@RKDA
003046  012777  000011  175740  MOV     #11,@RKCS       ;GO SEEK CYLINDER
                                        ;****WAIT FOR CONTROL READY IN FLAG MODE****
003054  005067  176030          CLR     WORK
003060  105777  175730          TSRDY:  TSTB    @RKCS           ;TEST FOR CONTROL READY
003064  100422                  BMI     ADT2X           ;BRANCH IF CONTROL READY
003066  005267  176016          INC     WORK            ;+1 COUNTER FOR READY
003072  001372                  BNE     TSRDY           ;TEST FOR READY AGAIN
003074  012777  000340  175700  MOV     #340,@PS        ;LOCKUP CPU
003102  012767  000004  175756  MOV     #4,ERCOUNT
003110  017767  175700  175772  MOV     @RKCS,WORK      ;FETCH CONTROL STATUS REG
003116  017767  175704  175766  MOV     @RKDS,WORK1     ;FETCH DRIVE STATUS REG
003124  004567  007272          ERR41:  JSR     %5,STAER1       ;REPORT ERROR
003130  000677                  BR      ADT2            ;LOOP ON ERROR
                                        ;*****WAIT FOR INTERRUPT FROM DRIVE READY*****
003132  005067  175752          ADT2X:  CLR     WORK
003136  052777  000100  175650  BIS     #BIT6,@RKCS     ;SET INTERRUPT ENABLE
003144  000777                  BR      .               ;WAIT FOR INTERRUPT WITH BRANCH .
003146  032777  000100  175652  BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003154  001017                  BNE     LPAD2           ;DRIVE READY OK!
003156  012777  000340  175616  MOV     #340,@PS        ;LOCK UP CPU
003164  012767  000005  175674  MOV     #5,ERCOUNT
003172  017767  175610  175710  MOV     @RKCS,WORK      ;FETCH CONTROL REG
003200  017767  175622  175704  MOV     @RKDS,WORK1     ;FETCH DRIVE STATUS
003206  004567  007210          ERR51:  JSR     %5,STAER1
003212  000646                  BR      ADT2            ;LOOP ON ERROR
003214  017767  175606  175666  LPAD21: MOV     @RKDS,WORK      ;FETCH DRIVE STATUS
003222  000241                  CLC
003224  006167  175660          ROL     WORK
003230  006167  175654          ROL     WORK
003234  006167  175650          ROL     WORK
003240  006167  175644          ROL     WORK
003244  042767  177770  175636  BIC     #177770,WORK
```

```
  503  003252  016767  175564  175632          MOV     FLAG,WORK1
  504  003260  006067  175626                  ROR     WORK1
  505  003264  006067  175622                  ROR     WORK1
  506  003270  042767  177770  175614          BIC     #177770,WORK1
  507  003276  026767  175610  175604          CMP     WORK1,WORK      ;IS THE POLE DRIVE NUMBER CORRECT
  508  003304  001407                          BEQ     POLOK           ;YES DRIVE NUMBER OK
  509  003306  012767  000006  175552  ERR61   MOV     #6,ERCOUNT      ;SET UP ERROR COUNT
  510  003314  004567  007102                  JSR     R5,STAER1       ;REPORT NUMBER INCORRECT
  511  003320  000167  177404                  JMP     AQT2            ;LOOP ON TEST
  512  003324  022767  014500  175520  POLOK:CMP     #14500,DMA
  513  003332  001404                          BEQ     .+12
  514  003334  062767  000040  175510          ADD     #40,DMA
  515  003342  000406                          BR      EXTAD2
  516  003344  032777  004000  175426          BIT     #BIT11,@SWR     ;CHECK TO LOOP ON TEST
  517  003352  001404                          BEQ     AQT2UX          ;GO TO NEXT TEST
  518  003354  000167  177350                  JMP     AQT2            ;LOOP ON TEST
  519  003360  000167  177350          EXTAD2: JMP     AQT2+4
```

```
                                        ;
                                        ;POLE TEST WITH ADDRESS VERIFICATION
                                        ;
                                        ;
                                        ;WRITE THE FIRST WORD IN SECTOR 0
                                        ;OF EACH CYLINDER WITH ITS SELF
                                        ;
                                        ;THEN POLE EACH CYLINDER AND CHECK
                                        ;THE ADDRESS
003364  005067  175462        AD?2UX: CLR     DMA
003370  012777  000001  175416        MOV     #BIT0,@RKCS     ;CLEAR CONTROL
003376  105777  175412        TSTB    @RKCS
003402  100375                BPL     .-4
003404  012706  001000        MOV     #1000,X6        ;SET UP STACK
003410  012777  005116  175416        MOV     #PWAIT,@VECTOR  ;SET UP INTERRUPT VECTOR
003416  012777  000340  175412        MOV     #340,@STATUS
003424  016777  175410  175350        MOV     PRIORITY,@PS
003432  012767  000001  175406        MOV     #1,WRQCT        ;SET UP WORD C
003440  012767  014750  175410        MOV     #OUTBUF,BUF
003446  016737  175400  014750  WRAUSK: MOV     DMA,@#OUTBUF
003454  104403                WRITE
003456  105777  175332        TSTB    @RKCS
003462  100375                BPL     .-4
003464  026727  175362  014500        CMP     DMA,#14500      ;IS IT THE LAST CYLINDER
003472  001404                BEQ     A2XSEK-4
003474  062767  000040  175350        ADD     #40,DMA
003502  000761                BR      WRAUSK
003504  005067  175342        CLR     DMA
003510  004567  004020        A2XSEK: JSR     X5,DSKNOS       ;SET UP DISK NO,
003514  056777  175370  175300        BIS     WORK,@RKDA      ;SET UP THE DRIVE
003522  042777  017777  175272        BIC     #17777,@RKDA
003530  052777  014500  175264        BIS     #14500,@RKDA    ;SET UP SEEK TO 300
003536  012777  000011  175250        MOV     #11,@RKCS       ;GO SEEK
003544  105777  175244        TSTB    @RKCS
003550  100375                BPL     .-4
003552  032777  000100  175246        BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003560  001774                BEQ     .-6
003562  042777  017777  175232        BIC     #17777,@RKDA
003570  056777  175256  175224        BIS     DMA,@RKDA
003576  012777  000011  175210        MOV     #11,@RKCS       ;GO SEEK CYLINDER
                        ;****WAIT FOR CONTROL READY IN FLAG MODE****
003604  005067  175300        CLR     WORK
003610  105777  175200        TS?AUX: TSTB    @RKCS           ;WAIT FOR CONTROL RDY
003614  100422                BMI     AUX2            ;CONTROL READY OK
003616  005267  175266        INC     WORK
003622  001372                BNE     TSTAUX          ;GO WAIT FOR CONTROL RDY
003624  012777  000340  175150        MOV     #340,@PS
003632  012767  000007  175226  ERR7:  MOV     #7,ERCOUNT
003640  017767  175150  175242        MOV     @RKCS,WORK      ;REPORT RKCS
003646  017767  175154  175236        MOV     @RKDS,WORK1     ;REPORT NO CONTROL READY
003654  004567  006476                JSR     X5,STAER        ;REPORT ERROR
003660  000713                BR      A2XSEK
```

```
                                        ;****WAIT FOR INTERRUPT FROM DRIVE READY****
003662  005067  175222          AUX21:  CLR     WORK
003666  052777  000100  175120          BIS     #BIT6,@RKCS     ;SET INTERRUPT ENABLE
003674  000777                          BR      .
003676  032777  000100  175122          BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003704  001017                          BNE     AU2LPA          ;DRIVE READY OK
```

```
                                                                                                                    
                                                                                                                    
003706  012777  000340  175066                MOV     #340,@PS
003714  012767  000010  175144  ERR10:        MOV     #10,ERCOUNT
003722  017767  175066  175160                MOV     @RKCS,WORK
003730  017767  175066  175154                MOV     @RKDA,WORK1
003736  004567  006460                        JSR     X5,STAER1       ;REPORT SEEK COMPLETE DRIVE NOT READY
003742  000662                                BR      A2XSEK
003744  012777  177777  175044  AU2LPA:       MOV     #-1,@RKWC
003752  016777  175130  175040                MOV     INBUF,@RKBA
003760  012777  000005  175026                MOV     #5,@RKCS        ;GO READ SECTOR
003766  105777  175022                        TSTB    @RKCS           ;CHECK FOR READY
003772  100375                                BPL     .-4
003774  026777  175052  175104                CMP     DMA,@INBUF      ;IS DATA CORRECT
004002  001414                                BEQ     LPAU2X
004004  016767  175042  175100                MOV     DMA,WORK1       ;WHAT SECTOR SHOULD CONTAIN
004012  017767  175070  175070                MOV     @INBUF,WORK     ;WHAT WAS IN SECTOR
004020  012767  000011  175040  ERR11:        MOV     #11,ERCOUNT
004026  004567  006370                        JSR     X5,STAER1       ;REPORT ERROR
004032  000626                                BR      A2XSEK
004034  022767  014500  175010  LPAU2X:       CMP     #14500,DMA      ;IS IT THE LAST CYLINDER
004042  001404                                BEQ     .+12            ;YES
004044  062767  000040  175000                ADD     #40,DMA         ;SET UP NEXT CYLINDER
004052  000616                                BR      A2XSEK          ;SEEK NEXT CYLINDER
004054  032777  004000  174716                BIT     #BIT11,@SWR     ;LOOP ON TEST?
004062  001402                                BEQ     .+6
004064  000167  177274                        JMP     AD?2UX          ;LOOP ON TEST
                                        ;
                                        ;DOLE TEST #3
                                        ;SEEK FROM 0 - 312 THEN
                                        ;SEEK FROM 312 - 0 THEN DECREMENT THE 312 COUNT
                                        ;AND INCREMENT 0 THEN GO AND SEEK AGAIN
                                        ;SEEK FROM 1 - 311 THEN
                                        ;SEEK FROM 311 - 1
                                        ;WHEN THE COUNTS HAVE GONE FROM 0 - 312
                                        ;AND 312 - 0 THE TEST IS DONE.
                                        ;
004070  005067  174774                AD?3UX: CLR     SAVE
004074  005067  175004                        CLR     PASSC
004100  012767  014500  174764                MOV     #14500,SAV1
004106  012777  000001  174700                MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
004114  105777  174674                        TSTB    @RKCS           ;WAIT FOR CONTROL READY
004120  100375                                BPL     .-4
004122  012706  001000                        MOV     #1000,X6        ;SET UP STACK
004126  012777  005116  174700                MOV     #PWAIT,@VECTOR  ;SET UP INTERRUPT VECTOR
004134  012777  000340  174674                MOV     #340,@STATUS
004142  016777  174672  174632                MOV     PRIORITY,@PS
004150  004567  004160                SEEKCY: JSR     X5,DSKNOS       ;SET UP DISK NUMBER
004154  056777  174730  174640                BIS     WORK,@RKDA      ;SET UP THE DRIVE NUMBER
004162  042777  017777  174632                BIC     #17777,@RKDA
004170  056777  174674  174624                BIS     SAVE,@RKDA      ;SET UP CYLINDER ADDRESS
004176  012777  000011  174610                MOV     #11,@RKCS       ;GO SEEK
                                        ;****WAIT FOR READY IN FLAG MODE****
004204  005067  174700                        CLR     WORK
```

```
                                                TSTRA3: TSTB    @RKCS           ;TEST FOR READY
004210  105777  174600                                BMI     NBOSL           ;READ SET OK
004214  100417                                        INC     WORK
004216  005267  174666                                BNE     TSTRA3          ;GO WAIT FOR READY
004222  001372                                        MOV     @RKCS,WORK1
004224  017767  174564  174660                        MOV     SAVE,WORK       ;CYLINDER #OF SEEK
004232  016767  174632  174650                ERR12:  MOV     #12,ERCOUNT
004240  012767  000012  174620                        JSR     X5,STAER1       ;REPORT CONTROL NOT RDY
004246  004567  006150                                BR      ADT3UX          ;LOOP ON TEST
004252  000706                                ;****WAIT FOR DRIVE READY INTERRUPT****
                                                NBOSL:  CLR     WORK
004254  005067  174630                                BIS     #BIT6,@RKCS     ;5 CT INTERRUPT ENABLE
004260  052777  000100  174526                        BR      .
004266  000777                                        JSR     X5,DSKNOS
004270  004567  004040                                MOV     WORK,@RKDA      ;SET UP THE DRIVE NUMBER
004274  056777  174610  174520                        BIC     #17777,@RKDA
004302  042777  017777  174512                        BIS     SAV1,@RKDA
004310  056777  174556  174504                        MOV     #11,@RKCS       ;GO SEEK
004316  012777  000011  174470                ;****WAIT FOR CONTROL READY IN FLAG MODE***
                                                        CLR     WORK
004324  005067  174560                        CHKRA3: TSTB    @RKCS
004330  105777  174460                                BMI     AISRY           ;READY SET OK
004334  100417                                        INC     WORK
004336  005267  174546                                BNE     CHKRA3          ;GO WAIT FOR FLAG
004342  001372                                        MOV     @RKCS,WORK1
004344  017767  174444  174540                        MOV     SAV1,WORK       ;CYLINDER # OF SEEK
004352  016767  174514  174530                ERR13:  MOV     #13,ERCOUNT     ;SET UP ERROR COUNT
004360  012767  000013  174500                        JSR     X5,STAER1       ;REPORT ERROR
004366  004567  006030                                BR      ADT3UX
004372  000636                                ;****WAIT FOR SEEK COMPLETE IN INTERRUPT MODE***
                                                AISRY:  CLR     WORK
004374  005067  174510                                BIS     #BIT6,@RKCS     ;SET INTERRUPT ENABLE
004400  052777  000100  174406                        BR      .
004406  000777
                                                ;
                                                ;
                                                ;
004410  005767  174470                                TST     PASSC           ;TEST FOR LAST PASS
004414  001003                                        BNE     .+10
004416  012767  177770  174460                        MOV     #-10,PASSC      ;SET UP NEW COUNT
004424  005267  174454                                INC     PASSC           ;+1 COUNT
004430  001011                                        BNE     LPAU3X-4
004432  005767  174434                                TST     SAV1
004436  001410                                        BEQ     LPAU3X
004440  062767  000040  174422                        ADD     #40,SAVE        ;INC, LOW CYLINDER ADDR, REG,
004446  162767  000040  174416                        SUB     #40,SAV1        ;DEC HI CYLINDER ADDR, REG,
004454  000167  177470                                JMP     SEEKCY          ;GO SEEK
004460  032777  004000  174312                LPAU3X: BIT     #BIT11,@SWR     ;LOOP ON TEST
004466  001200                                        BNE     ADT3UX          ;YES LOOP ON TEST
004470  000400                                        BR      SAWTH
                                                ;THIS TEST IS USED TO GENERATE NOISE
                                                ;IN DRIVE ADDRESS CIRCUIT
                                                ;THE ROUTINE SEEKS TO CYLINDER N
```

```
                                        ;THEN TO CYLINDER N+2
                                        ;THEN TO CYLINDER N+1
                                        ;THEN TO CYLINDER N+3
                                        ;THIS GENERATES A SAW TOOTH AFFECT WHEN
                                        ;SEEKING,
                                        ;
                                        ;
004472  012767  177766  174404  SAWTH1  MOV     #.-10,;PASSC
004500  012706  001000                  MOV     #1000,%6        ;SET UP STACK
004504  012777  005116  174322          MOV     #PWAIT,@VECTOR  ;SET UP DISK VECTOR
004512  012777  000340  174316          MOV     #340,@STATUS
004520  016777  174314  174254          MOV     PRIORITY,@#S    ;SET UP DRIVE #
004526  012767  000100  174334          MOV     #100,SAVE
004534  004567  003574          NOISET1 JSR     %5,DSKNOS       ;SET UP CPU STATUS
004540  056777  174344  174254          BIS     WORK,@RKDA      ;SET THE DRIVE NUMBER
004546  042777  017777  174246          BIC     #17777,@RKDA
004554  056777  174310  174240          BIS     SAVE,@RKDA      ;SET FOR FIRST SEEK
004562  012777  000011  174224          MOV     #11,@RKCS       ;GO SEEK
004570  105777  174220                  TSTB    @RKCS           ;TEST FOR READY
004574  100375                          BPL     .-4             ;NO FLAG GO BACK
004576  052777  000100  174210          BIS     #BIT6,@RKCS     ;SET INT ENABLE
004604  000777                          BR      .               ;WAIT FOR INTERRUPT FROM SCP
004606  042777  000100  174200          BIC     #BIT6,@RKCS     ;CLEAR INTERRUPT ENABLE
004614  005777  174174                  TST     @RKCS           ;TEST FOR ERRORS
004620  100015                          BPL     SAW1            ;NO ERRORS GO ON
004622  017767  174166  174262          MOV     @RKCS,WORK1     ;FETCH RKCS
004630  017767  174166  174252          MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS
004636  012767  000014  174222  ERR14I  MOV     #14,ERCOUNT
004644  004567  005552                  JSR     %5,STAER1       ;REPORT ERROR
004650  000167  177616                  JMP     SAWTH           ;LOOP ON ERROR
004654  162777  000040  174140  SAW1I   SUB     #40,@RKDA       ;DEC, CYLINDER NO,
004662  012777  000011  174124          MOV     #11,@RKCS       ;GO SEEK
004670  105777  174120                  TSTB    @RKCS           ;TEST FOR CONTROL READY
004674  100375                          BPL     .-4             ;NO FLAG GO BACK
004676  052777  000100  174110          BIS     #BIT6,@RKCS     ;SET INT, ENABLE
004704  000777                          BR      .               ;WAIT FOR INT FROM SCP
004706  042777  000100  174100          BIC     #BIT6,@RKCS     ;CLEAR INT ENABLE
004714  005777  174074                  TST     @RKCS           ;TEST FOR ERRORS
004720  100015                          BPL     SAW2            ;BRANCH IF NO ERRORS
004722  017767  174066  174162          MOV     @RKCS,WORK1     ;FETCH FUNCTION REGISTER
004730  017767  174066  174152          MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS REGISTER
004736  012767  000015  174122  ERR15I  MOV     #15,ERCOUNT
004744  004567  005452                  JSR     %5,STAER1       ;REPORT ERROR
004750  000167  177516                  JMP     SAWTH           ;LOOP ON ERROR
004754  062777  000100  174040  SAW2I   ADD     #100,@RKDA      ;SET UP NEXT CYLINDER
004762  012777  000011  174024          MOV     #11,@RKCS       ;GO SEEK
004770  105777  174020                  TSTB    @RKCS           ;TEST FOR READY
004774  100375                          BPL     .-4             ;GO BACK NO FLAG
004776  052777  000100  174010          BIS     #BIT6,@RKCS     ;SET INT,ENABLE
005004  000777                          BR      .               ;WAIT FOR INT, FROM SCP
005006  042777  000100  174000          BIC     #BIT6,@RKCS     ;CLR INT, ENABLE
005014  005777  173774                  TST     @RKCS           ;TEST FOR ERRORS
005020  100015                          BPL     SAW3            ;BRANCH IF NO ERRORS
005022  017767  173766  174062          MOV     @RKCS,WORK1     ;FETCH FUNCTION REGISTER
```

```
  742  005030  017767  173766  174052          MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS
  743  005036  012767  000016  174022  ERR16:  MOV     #16,ERCOUNT
  744  005044  004567  005352                  JSR     X5,STAER1       ;REPORT ERRO
  745  005050  000167  177416                  JMP     SAWTH           ;LOOP ON ERROR
  746  005054  062767  000140  174006  SAW3:   ADD     #140,SAVE       ;SET UP NEXT SAW TOOTH
  747  005062  022767  014540  174000          CMP     #14540,SAVE     ;IS IT THE LAST SAW TOOTH
  748  005070  001221                          BNE     NOISET
  749  005072  032777  004000  173700          BIT     #BIT11,@SWR     ;CHECK TO LOOP ON TEST
  750  005100  001402                          BEQ     .+6
  751  005102  000167  177364                  JMP     SAWTH           ;LOOP ON TEST
  752  005106  005267  173772                  INC     PASSC
  753  005112  001205                          BNE     NOISET+6
  754  005114  000413                          BR      ADT3
  755                                  ;
  756                                  ;
  757                                  ;
  758                                  ;
  759  005116  032777  020000  173670  PWAIT:  BIT     #BIT13,@RKCS    ;TEST FOR SEEK COMPLETE
  760  005124  001004                          BNE     XWPAT           ;SEEK NOT COMPLETE
  761  005126  005267  173756                  INC     WORK
  762  005132  001401                          BEQ     .+4
  763  005134  000002                          RTI                     ;RETURN TO ADT2X SEEK COMPLETE
  764  005136  062716  000002          XWPAT:  ADD     #2,(6)          ;SEEK NOT COMPLETE
  765  005142  000002                          RTI                     ;RETURN TO WAIT
  766                                  ;
  767                                  ;
```

```
768                                     ;
769                                     ;THIS IS A LOOK AHEAD TEST ,
770                                     ;IT FIRST LOOKS FOR ADDRESS ZERO BY READING THE RKDS REGISTER
771                                     ;THEN CONTINUING TO READ THE RKDS REGISTER IT LOOKS FOR AN ADRESS
772                                     ;WHICH IS EQUAL TO OR ONE GREATER THAN THE PREVIOUS,
773                                     ;
774                                     ;
775                                     ;;
776  005144  012706  001000            ADDR31: MOV     #1000,X6        ;SET UP STACK
777  005150  012777  000001  173636            MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
778  005156  105777  173632                    TSTB    @RKCS
779  005162  100375                            BPL     .-4
780  005164  004567  003144                    JSR     X5,DSKNOS       ;SET THE DISK NUMBER
781  005170  056777  173714  173624            BIS     WORK,@RKDA      ;SET THE DRIVE NUMBER
782  005176  005067  173666                    CLR     SAVE
783  005202  005067  173704                    CLR     WORK1
784  005206  017767  173614  173674  TSTZER: MOV     @RKDS,WORK
785  005214  032767  000400  173666            BIT     #BIT8,WORK      ;CHECK DRIVE STATUS OK
786  005222  001771                            BEQ     TSTZER
787  005224  032767  000017  173656            BIT     #17,WORK        ;TEST FOR SECTOR 0
788  005232  001413                            BEQ     TSTNXS
789  005234  005267  173652                    INC     WORK1           ;WAIT FOR SECTOR ZERO
790  005240  005767  173646                    TST     WORK1
791  005244  001360                            BNE     TSTZER
792  005246  012767  000017  173612            MOV     #17,ERCOUNT     ;COULD NOT FIND SECTOR 0
793  005254  004567  005076            ERR17:  JSR     X5,STAER        ;REPORT ERROR
794  005260  000731                            BR      ADT3            ;LOOP ON ERROR
795  005262  017767  173540  173620  TSTNXS: MOV     @RKDS,WORK
796  005270  032777  000400  173530            BIT     #BIT8,@RKDS     ;TST FOR DRIVE STATUS OK
797  005276  001771                            BEQ     TSTNXS
798  005300  042767  177760  173602            BIC     #177760,WORK
799  005306  016767  173576  173576            MOV     WORK,WORK1
800  005314  026767  173550  173570            CMP     SAVE,WORK1
801  005322  001757                            BEQ     TSTNXS          ;IT WAS THE SAME ADDRESS
802  005324  166767  173540  173560            SUB     SAVE,WORK1
803  005332  022767  000001  173552            CMP     #1,WORK1                ;WAS THE ADDR, ONE GREATER
804  005340  001007                            BNE     LKADER          ;ADDR GREATER THAN +1
805  005342  022767  000012  173520            CMP     #12,SAVE        ;IS IT THE LAST SECTOR
806  005350  001414                            BEQ     LPAD3           ;YES EXIT
807  005352  005267  173512                    INC     SAVE
808  005356  000741                            BR      TSTNXS          ;CHECK NEXT SECTOR
809  005360  012767  000020  173500  LKADER: MOV     #20,ERCOUNT
810  005366  016767  173476  173514            MOV     SAVE,WORK
811  005374  004567  005022            ERR20:  JSR     X5,STAER1
812  005400  000661                            BR      ADT3
813  005402  000240                    LPAD3:  NOP
```

```
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
005404  005067  173442          ;DATATXI CLR     DMA
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
005410  016767  173450  173430  ;DATATI  MOV     SWRDCT,WRDCT
005416  012777  006770  173410  ;        MOV     #UKINT,@VECTOR
005424  012777  000340  173404  ;        MOV     #340,@STATUS
005432  012767  005452  173440  ;        MOV     #LDAT,WRDER     ;SETUP FOR HARD ERROR
005440  012777  000340  173334  ;        MOV     #340,@PS        ;LOCK UP PROCESSOR PRIORITY
005446  004567  003100          ;        JSR     %5,PASEL        ;SET UP DATA BUFFERS
005452  004567  002062          ;LDATI   JSR     %5,OPDSEL       ;SET UP DISK ADDRESS
005456  032767  040000  173356  ;        BIT     #BIT14,FLAG     ;TEST FOR WRITE
005464  001421                  ;        BEQ     ESH             ;TEST FOR READ
005466  012767  014750  173362  ;        MOV     #OUTBUF,BUF     ;SETUP OUTPUT BUFFER
005474  104503                  ;        WRITE   +100            ;WRITE WITH INT, ENABLED
005476  032777  001000  173274  ;        BIT     #BIT9,@SWR      ;FIND OUT HOW TO WAIT FOR INT
005504  001002                  ;        BNE     WRWAIT          ;WAIT WITH WAIT INSTRUCTION
005506  000004                  ;        IOT                     ;WAIT IN BACKGROUND TEST
005510  000404                  ;        BR      XSLH
005512  016777  173322  173262  ;WRWAITI MOV     PRIORITY,@PS
005520  000001                  ;        WAIT                    ;WAIT FOR FLAG
005522  004767  002076          ;XSLHI   JSR     %7,DISBUF       ;SET BUFFER FOR WRITE
005526  000751                  ;        BR      LDAT
005530  004567  002004          ;ESHI    JSR     %5,OPUSEL       ;OPERATOR SELECTED CYLINDER?
005534  032767  010000  173300  ;        BIT     #BIT12,FLAG     ;TEST FOR READ
005542  001451                  ;        BEQ     RUCKLP
005544  016767  173336  173304  ;        MOV     INBUF,BUF       ;SETUP OUTPUT BUFFER
005552  042767  000003  173262  ;        BIC     #3,FLAG         ;CLEAR RE-READ COUNT
```

```
850
851  005560  012777  000340  173214  DSKRD1: MOV     #340,@PS
852  005566  005267  173250                  INC     FLAG
853  005572  104505                         READ    +100                    ;READ & INT ENABLE
854  005574  032777  001000  173176         BIT     #BIT9,@SWR              ;FIND OUT HOW TO WAIT FOR INT,
855  005602  001002                         BNE     RDWAIT                  ;WAIT WITH WAIT INSTRUCTION
856  005604  000004                         IOT
857  005606  000404                         BR      ELH
858  005610  016777  173224  173164  RDWAIT: MOV     PRIORITY,@PS            ;SET UP PRIORITY
859  005616  000001                         WAIT                            ;WAIT FOR FLAG
860  005620  032777  010000  173152  ELH:   BIT     #BIT12,@SWR
861  005626  001002                         BNE     ADRD
862  005630  004567  003204                JSR     X5,COMPARE              ;COMPARE OUTBUFFER TO INBUFFER
863  005634  016767  173202  173246  ADRD:  MOV     FLAG,WORK               ;CHECK DISK RE-READ COUNT
864  005642  042767  177774  173240         BIC     #177774,WORK            ;DO 3 RE-READS;
865  005650  022767  000003  173232         CMP     #3,WORK
866  005656  001340                         BNE     DSKRD                   ;DO ANOTHER RE-READ
867  005660  004767  001740                JSR     X7,DISBUF               ;GO SET UP DISK BUFFER;
868  005664  000721                         BR      ESH
869  005666  004567  001646  RDCKLP: JSR     X5,OPQSEL
870  005672  032767  000100  173142         BIT     #BIT6,FLAG
871  005700  001421                         BEQ     SLH
872  005702  012777  000340  173072         MOV     #340,@PS
873  005710  104513                         RDCK    +100
874  005712  032777  001000  173060         BIT     #BIT9,@SWR
875  005720  001002                         BNE     RCHWAIT
876  005722  000004                         IOT
877  005724  000404                         BR      ESRDCH
878  005726  016777  173106  173046  RCHWAIT: MOV    PRIORITY,@PS
879  005734  000001                         WAIT
880  005736  004767  001662          ESRDCH: JSR     X7,DISBUF
881  005742  000751                         BR      RDCKLP
882  005744  004567  001570          SLH:   JSR     X5,OPQSEL               ;IS THE OPERATOR SELECTING THE TRACK
883  005750  032767  020000  173064         BIT     #BIT13,FLAG             ;TEST FOR WRITE CHECK
884  005756  001424                         BEQ     MSTR
885  005760  012767  014750  173070         MOV     #OUTBUF,BUF             ;SET UP CURRENT ADDRESS
886  005766  012777  000340  173006         MOV     #340,@PS
887  005774  104507                 WRCHECK +100
888  005776  032777  001000  172774         BIT     #BIT9,@SWR
889  006004  001002                         BNE     WCWAIT                  ;WAIT FOR FLAG IN WAIT INST,
890  006006  000004                         IOT                             ;WAIT FOR FLAG IN BACKGROUND TEST
891  006010  000404                         BR      XESH
892  006012  016777  173022  172762  WCWAIT: MOV     PRIORITY,@PS
893  006020  000001                         WAIT                            ;WAIT FOR THE FLAG
894  006022  004767  001576          XESH:  JSR     X7,DISBUF               ;SET UP THE DISK BUFFER
895  006026  000746                         BR      SLH
896  006030  032777  000400  172742  MSTR:  BIT     #BIT8,@SWR              ;ARE WE LOOPING ON AN ADDRESS
897  006036  001402                         BEQ     .+6                     ;NO!ADDRESS SELECTION UNDER PROGRAM CONT
898  006040  000167  177344                JMP     DATAT                   ;LOOP ON ADDRESS
899  006044  005767  172772                TST     FLAG
900  006050  100002                         BPL     .+6                     ;UNDER PROGRAM CONTROL
901  006052  000167  000452                JMP     CHKDOS                  ;OPERATOR SELECTED PATTERN
902  006056  062767  000002  172770         ADD     #2,PATNU                ;INC PATTERN INDEX
903  006064  022767  000044  172762         CMP     #44,PATNU
```

```
 904  006072  001402                          BEQ     .+6
 905  006074  000167  177310                   JMP     DATAT           ;NOT LAST PATTERN EXIT
 906  006100  005067  172750                   CLR     PATNU           ;LAST PATTERN EXIT
 907                                  ;THIS IS A RANDOM ADDRESS, RANDOM DATA TEST
 908                                  ;
 909                                  ;
 910                                  ;
 911  006104  012767  177000  172772  RANEX1: MOV     #-1000,PASSC    ;SET UP PASS COUNT
 912  006112  042767  001003  172722          BIC     #1003,FLAG      ;CLEAR ERROR FLAG BITS
 913  006120  012706  001000                  MOV     #1000,%6        ;SET UP STACK
 914  006124  012767  006416  172746          MOV     #RANER,HRDER    ;SET UP FOR HARD ERROR
 915  006132  012777  006770  172674          MOV     #DKINT,@VECTOR  ;SET UP INTERRUPT VECTOR
 916  006140  012777  000340  172670          MOV     #340,@STATUS
 917  006146  016777  172666  172626          MOV     PRIORITY,@PS    ;SET PRIORITY TO LEVEL 5
 918  006154  012767  000001  172726  WRLG1:  MOV     #1,WORK         ;SET UP RANDOM GENERATOR WORD
 919  006162  012701  014750                  MOV     #OUTBUF,%1
 920  006166  004567  002452                  JSR     %5,RANDOM       ;GENERATE RANDOM DATA
 921  006172  016767  006552  172652          MOV     OUTBUF,DMA      ;SET UP DISK ADDRESS
 922  006200  042767  160000  172644          BIC     #160000,DMA
 923  006206  022767  014533  172636          CMP     #14533,DMA
 924  006214  101757                          BLOS    WRLG
 925  006216  042767  177760  006524          BIC     #177760,OUTBUF
 926  006224  022767  000014  006516          CMP     #14,OUTBUF      ;IS IT THE LAST SECTOR
 927  006232  101750                          BLOS    WRLG            ;BRALCH IF YES
 928  006234  012767  000200  172604          MOV     #200,WRDCT      ;SET UP WOPD COUNT =1SECTOP
 929  006242  016767  172600  172640          MOV     WRDCT,WORK      ;GENERATE RANDOM BUFFER
 930  006250  012701  014750                  MOV     #OUTBUF,%1
 931  006254  004567  002364                  JSR     %5,RANDOM
 932  006260  012777  000001  172526          MOV     #BIT0,@RKCS     ;CLEARTHE CONTROL
 933  006266  105777  172522                  TSTB    @RKCS           ;IS THE CONTROL READY
 934  006272  100375                          BPL     .-4
 935  006274  012767  014750  172554          MOV     #OUTBUF,BUF     ;SET UP OUTPUT BUFFER
 936  006302  104503                          WRITE   +100    )
 937  006304  032777  001000  172466          BIT     #BIT9,@SWR
 938  006312  001002                          BNE     .+6
 939  006314  000004                          IOT
 940  006316  000404                          BR      .+12
 941  006320  016777  172514  172454          MOV     PRIORITY,@PS    ;LOCK UP CPU
 942  006326  000001                          WAIT
 943  006330  104507                          WRCHECK +100            ;WRITE CHECK THE DISK
 944  006332  032777  001000  172440          BIT     #BIT9,@SWR
 945  006340  001002                          BNE     .+6
 946  006342  000004                          IOT
 947  006344  000404                          BR      .+12
 948  006346  016777  172466  172426          MOV     PRIORITY,@PS
 949  006354  000001                          WAIT
 950  006356  016767  172524  172472          MOV     INBUF,BUF
 951  006364  104505                          READ    +100
 952  006366  032777  001000  172404          BIT     #BIT9,@SWR
 953  006374  001002                          BNE     .+6
 954  006376  000004                          IOT
 955  006400  000404                          BR      .+12
 956  006402  016777  172432  172372          MOV     PRIORITY,@PS
 957  006410  000001                          WAIT
```

```
                                                JSR     X5,COMPARE      ;COMPARE THE DATA FOR ERRORS
006412  004567  002422
006416  032767  001000  172416  RANER: BIT      #BIT9,FLAG      ;TEST FOR ERRORS
006424  001422                          BEQ     EXRAX           ;NO ERRORS
006426  042767  001000  172406          BIC     #BIT9,FLAG
006434  004567  004312                  JSR     X5,CONV         ;SET UP FOR ADDRESS REPORT
006440  001052                          DMA
006442  013613                          MES1
006444  000006                          6
006446  004567  004300                  JSR     X5,CONV         ;SET UP WORD COUNT FOR REPORT
006452  001046                          WRDCT
006454  013677                          MES3
006456  000006                          6
006460  104000                          EMT+0
006462  013575                          MED6
006464  013613                          MES1
006466  013677                          MES3
006470  177777                          -1
006472  005267  172406          EXRAX: INC PASSC                ;+1 PASS COUNT
006476  001226                          BNE     WRLG            ;BRANCH IF TEST NNT OVER
006500  032777  004000  172272          BIT     #BIT11,@SWR     ;LOOP ON TEST
006506  001402                          BEQ     .+6
006510  000167  177370                  JMP     RANEX           ;YES LOOP ON TEST
                                        )
                                        ;CHECK FOR MULTI DISK MODE
                                        ;IF IN MULTI DISK MODE REPORT "END"
                                        ;IF LAST DISK ON SYSTEM HAS BEEN
                                        ;EXERCISED;
                                        )
                                        )
006514  005067  172332                  CLR     DMA
006520  032767  004000  172314          BIT     #BIT11,FLAG     ;ARE WE IN MULTI DISK MODE
006526  001426                          BEQ     REPOEN          ;REPORT "END"
006530  016700  172306          CHKDOS: MOV     FLAG,X0         ;WHAT DISK ARE WE ON
006534  042700  177743                  BIC     #177743,X0      ;IF LAST DISK ON SYSTEM
006540  020067  172332                  CMP     X0,DSKNOR       ;REPORT END
006544  001417                          BEQ     REPOEN          ;REPORT "END" LAST DISK
006546  010003                          MOV     X0,X3
006550  000241                          CLC
006552  006003                          ROR     X3
006554  022763  000020  013450          CMP     #20,TABLE(3)
006562  101004                          BHI     DRVEROK
006564  062767  000004  172250          ADD     #4,FLAG
006572  000756                          BR      CHKDOS
006574  062767  000004  172240  DRVEROK:ADD     #4,FLAG         ;INC, DISK NO,
006602  000414                          BR      EXTPP           ;EXERCISE DISK
006604  104001                  REPOEN: EMT     +1
006606  014667                          END                     ;REPORT END OF PASS
006610  042767  000034  172224          BIC     #34,FLAG        ;RESET DRIVE NUMBER TO ZERO
006616  013700  000042                  MOV     @#42,X0         ;WAS THIS A MONITOR LOAD?
006622  001404                          BEQ     EXTPP           ;NO - RECYCLE
006624  004710                  LOGICAL:JSR     X7,(0)
006626  000240                          NOP
006630  000240                          NOP
006632  000240                          NOP
```

```
1012  006634  005077  172162          EXTPPI  CLR     @RKDA
1013  006640  000005                          RESET
1014  006642  000167  173324                  JMP     ADTST           ;RECYCLE
1015                                  ;
1016                                  ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
1017                                  ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
1018                                  ;BYTE OF THE CONTROL REGISTER.
1019                                  ;
1020  006646  016705  172150          DISK1   MOV     RKDA,R5         ;SET UP TO LOAD DISK REG
1021  006652  016715  172174                  MOV     DMA,(5)         ;LOAD WORD ADDRESS
1022  006656  016767  172160  172224          MOV     FLAG,WORK
1023  006664  000241                          CLC                     ;SET UP THE DRIVE NUMBER
1024  006666  006067  172216                  ROR     WORK
1025  006672  006067  172212                  ROR     WORK
1026  006676  006067  172206                  ROR     WORK
1027  006702  006067  172202                  ROR     WORK
1028  006706  006067  172176                  ROR     WORK
1029  006712  006067  172172                  ROR     WORK
1030  006716  042767  017777  172164          BIC     #17777,WORK
1031  006724  056715  172160                  BIS     WORK,(5)
1032  006730  016745  172122                  MOV     BUF,-(5)        ;SET UP CURRENT ADDRESS
1033  006734  016745  172106                  MOV     WRDCT,-(5)      ;LOAD WORD COUNT
1034  006740  005415                          NEG     (5)
1035  006742  011604                          MOV     (6),R4
1036  006744  014467  172140                  MOV     -(4),WORK       ;
1037  006750  042767  177600  172132          BIC     #177600,WORK
1038  006756  016745  172126                  MOV     WORK,-(5)       ;LOAD FUNCTION REG.
1039  006762  052715  000400                  BIS     #BIT8,(5)       ;SET STOP ON SOFT ERROR
1040  006766  000002                          RTI                     ;RETURN FROM TRAP
1041                                  ;
1042                                  ;
1043                                  ;
1044                                  ;
1045                                  ;
1046                                  ;
1047                                  ;
1048                                  ;RK11 DISK INTERRUPT HANDLER
1049                                  ;ROUTINE CONTINUES ON ERRORS
1050  006770  005046                  DKINT:  CLR     -(6)            ;CLEAR STACK
1051  006772  012746  007000                  MOV     #1S,-(6)        ;SET RETURN
1052  006776  000002                          RTI                     ;CLEAR TBIT
1053  007000  005777  172010          1S:     TST     @RKCS
1054                                  ;
1055  007004  100402                          BMI     .+6
1056  007006  000167  000476                  JMP     INTEXT          ;JUMP IF NO ERRORS
1057  007012  004767  004410                  JSR     R7,INCTAB
1058  007016  052767  001000  172016          BIS     #BIT9,FLAG      ;SET ERROR BIT
1059  007024  017767  171774  172056          MOV     @RKER,WORK      ;REPORT ERROR
1060  007032  004567  003714                  JSR     R5,CONV         ;CONVERT TO ASCII
1061  007036  001110                          WORK
1062  007040  013630                          MES1A
1063  007042  000006                          6
1064  007044  104001                          EMT+1
1065  007046  013505                          HED2
```

```
1066   007050   032777   000002   171736          BIT     #BIT1,@RKCS     ;CHECK FOR READ
1067   007056   001015                            BNE     DELMES          ;IF READING REPORT WHICH READ
1068   007060   016767   171756   172022          MOV     FLAG,WORK
1069   007066   042767   177774   172014          BIC     #177774,WORK
1070   007074   004567   003652                   JSR     %5,CONV
1071   007100   001110                            WORK
1072   007102   014020                            MES13
1073   007104   000001                            1
1074   007106   104001                            EMT+1
1075   007110   014020                            MES13
```

```
1076
1077  007112  005777  171706          DELMES1 TST     @RKER           ;TEST FOR DRIVE ERROR
1078  007116  100012                          BPL     REDA            ;BRANCH NO DRIVE ERROR
1079  007120  017767  171702  171762          MOV     @RKDS,WORK      ;FETCH DRIVE STATUS
1080  007126  004567  003620                  JSR     %5,CONV
1081  007132  001110                          WORK
1082  007134  013662                          MES2A
1083  007136  000006                          6
1084  007140  104001                          EMT+1
1085  007142  013662                          MES2A
1086  007144  017767  171652  171736  REDA1   MOV     @RKDA,WORK      ;SET UP 16 BITS OF ADDR,
1087                                  ;ALTER ADDRESS FOR CORRECT ERROR REPORT
1088  007152  032777  040000  171634          BIT     #BIT14,@RKCS    ;WAS IT A DRIVE ERROR
1089  007160  001036                          BNE     ADRREP          ;YES DO NOT ALTER ADDRESS
1090  007162  016767  171722  171722          MOV     WORK,WORK1
1091  007170  042767  177760  171714          BIC     #177760,WORK1   ;MASK SECTOR
1092  007176  032767  000017  171706          BIT     #17,WORK1       ;TEST FOR ERROR IN LAST SECTOR
1093  007204  001403                          BEQ     .+10            ;ERROR IN LAST SECTOR
1094  007206  005367  171676                  DEC     WORK            ;ERROR NOT IN LAST SECTOR
1095  007212  000421                          BR      ADRREP          ;REPORT ERROR
1096  007214  052767  000013  171666          BIS     #13,WORK        ;SET UP FOR ERROR IN LAST SECTOR
1097  007222  032767  000020  171660          BIT     #BIT4,WORK      ;TEST FOR WHICH SURFACE
1098  007230  001404                          BEQ     .+12            ;ERROR ON SURFACE ONE
1099  007232  042767  000020  171650          BIC     #BIT4,WORK      ;ERROR ON SURFACE ZERO
1100  007240  000406                          BR      ADRREP          ;REPORT ERROR
1101  007242  052767  000020  171640          BIS     #BIT4,WORK      ;ERROR ON SURFACE ONE
1102  007250  162767  000040  171632          SUB     #40,WORK        ;SUB, FROM CYLINDER NUMBER
1103  007256  004567  003470          ADRREP1 JSR     %5,CONV         ;CONVERT TO ASCII
1104  007262  001110                          WORK
1105  007264  013613                          MES1
1106  007266  000006                          6
1107  007270  104000                          EMT+0
1108  007272  013630                          MES1A
1109  007274  013613                          MES1
1110  007276  177777                          -1
1111  007300  017767  171510  171602          MOV     @RKCS,WORK      ;SET UP STATUS
1112  007306  004567  003440                  JSR     %5,CONV
1113  007312  001110                          WORK
1114  007314  013645                          MES2
1115  007316  000006                          6
1116  007320  104001                          EMT+1
1117  007322  013645                          MES2
1118  007324  032777  002000  171474          BIT     #BIT10,@RKDS    ;TEST FOR DRIVE UNSAFE
1119  007332  001402                          BEQ     .+6             ;DRIVE SAFE CONTINUE
1120  007334  000000                          HALT                    ;DRIVE UNSAFE HALT
1121  007336  000777                          BR      .               ;LOCK UP TEST
1122  007340  032777  040000  171446  CKHRDER1BIT     #BIT14,@RKCS    ;TEST FOR HARD ERROR
1123  007346  001436                          BEQ     SOFTER          ;GO AND CONTINUE SOFT ERROR
1124  007350  012706  001000          HRDEXIT1MOV     #1000,%6
1125  007354  032777  002000  171416          BIT     #BIT10,@SWR     ;HALT ON ERROR
1126  007362  001401                          BEQ     .+4
1127  007364  000000                          HALT                    ;YES HALT BIT 10 SET IN SR
1128  007366  012777  000001  171420          MOV     #BIT0,@RKCS
1129  007374  105777  171414                  TSTB    @RKCS
```

```
1130  007400  100375                        BPL     .-4
1131  007402  004567  000726                JSR     R5,DSKNOS       ;SET UP THE DRIVE NUMBER
1132  007406  012777  000015  171400        MOV     #15,@RKCS       ;RESTORE DRIVE
1133  007414  032777  000100  171404        BIT     #BIT6,@RKDS     ;IS THE SEARCH COMPLETE
1134  007422  001774                        BEQ     .-6
1135  007424  012777  000001  171362        MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
1136  007432  105777  171356                TSTB    @RKCS           ;IS THE CONTROL READY
1137  007436  100375                        BPL     .-4
1138  007440  000177  171434                JMP     @HRDER          ;EXIT HARD ERROR
1139  007444  005777  171346        SOFTER: TST     @RKWC           ;CHECK FOR X-FER DONE
1140  007450  001417                        BEQ     INTEXT          ;EXIT FROM ROUTINE
1141  007452  032777  001000  171320        BIT     #BIT9,@SWR
1142  007460  001402                        BEQ     .+6
1143  007462  162716  000002                SUB     #2,(6)          ;X-FER NOT DONE SET UP FOR RETURN
1144  007466  032777  002000  171304        BIT     #BIT10,@SWR     ;HALT ON ERROR
1145  007474  001401                        BEQ     .+4
1146  007476  000000                        HALT                    ;YES HALT BIT10 SET IN SR
1147  007500  052777  000001  171306        BIS     #BIT0,@RKCS     ;SET GO AND CONTINUE
1148  007506  000002                        RTI                     ;RETURN TO WAIT INSTR;
1149                                        ;
1150                                        ;
1151  007510  032777  020000  171262  INTEXT: BIT    #BIT13,@SWR     ;HALT ON COMPLETION FLAG
1152  007516  001401                        BEQ     .+4
1153  007520  000000                        HALT                    ;YES BIT 13 SET IN SR HALT
1154  007522  032777  001000  171250        BIT     #BIT9,@SWR
1155  007530  001002                        BNE     .+6
1156  007532  012706  000774                MOV     #774,R0         ;RESET STACK
1157  007536  000002                        RTI                     ;EXIT
```

```
1158
1159                                    ;ROUTINE TO SET UP CYLINDER # FROM SR
1160                                    ;CYLINDER NO, IN SR 7-0
1161                                    ;ENTER FROM JSR X5, OPOSEL
1162                                    ;
1163  007540  032767  000040  171274  OPOSEL: BIT     #BIT5,FLAG      ;DOES SR CONTAIN CYLINDER #
1164  007546  001001                          BNE     .+4
1165  007550  000205                          RTS     X5              ;NO: EXIT
1166  007552  017767  171222  171330          MOV     @SWR,WORK       ;FETCH SR
1167  007560  042767  177400  171322          BIC     #177400,WORK    ;MASK CYLINDER BITS
1168  007566  000241                          CLC
1169  007570  006167  171314                  ROL     WORK
1170  007574  006167  171310                  ROL     WORK
1171  007600  006167  171304                  ROL     WORK
1172  007604  006167  171300                  ROL     WORK
1173  007610  006167  171274                  ROL     WORK
1174  007614  116767  171270  171230          MOVB    WORK,@MA        ;LOAD CYLINDER BITS
1175  007622  000205                          RTS     X5              ;EXIT
```

```
1176
1177                                    ;
1178                                    ;ROUTINE TO SETUP DISK BUFFERS
1179                                    ;ADD WORD COUNT TO STARTING DISK ADDRESSES
1180                                    ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
1181                                    ;
1182  007624  032777  000400  171146  DISBUF: BIT     #BIT8,@SWR
1183  007632  001401                          BEQ     .+4
1184  007634  000510                          BR      BUFEXIT+4       ;LOOPING ON A SELECTED ADDRESS
1185  007636  004767  000542                  JSR     %7,BLSZ         ;DEFINE BLOCK SIZE
1186  007642  016767  171234  171242          MOV     BLOCK,WORK1
1187  007650  016767  171176  171232  INCSEC: MOV     DMA,WORK
1188  007656  042767  177700  171224          BIC     #177700,WORK    ;MASK OUT SECTOR COUNT
1189  007664  022767  000013  171216          CMP     #13,WORK        ;CHECK FOR LAST SECTOR
1190  007672  001406                          BEQ     INCSUR          ;CHECK SURFACE
1191  007674  005267  171152                  INC     DMA             ;+1 SECTOR COUNT
1192  007700  005367  171170          DECBLK: DEC     BLOCK           ;-1 FROM BLOCK COUNT
1193  007704  001423                          BEQ     COMDAR          ;CMP DMA TO RKDA
1194  007706  000760                          BR      INCSEC          ;RECYCLE
1195  007710  042767  000017  171134  INCSUR: BIC     #17,DMA         ;FETCH ADDRESS
1196  007716  032767  000020  171126          BIT     #BIT4,DMA       ;CHECK FOR SURFACE
1197  007724  001407                          BEQ     SWSUR           ;+1 SURFACE
1198  007726  062767  000040  171116          ADD     #40,DMA         ;INC, CYLINDER NUMBER
1199  007734  042767  000020  171110          BIC     #BIT4,DMA       ;CLEAR SURFACE
1200  007742  000756                          BR      DECBLK          ;RECYCLE
1201  007744  052767  000020  171100  SWSUR:  BIS     #BIT4,DMA
1202  007752  000752                          BR      DECBLK
1203  007754  004567  000354          COMDAR: JSR     %5,DSKNOS       ;SET UP THE DRIVE NUMBER
1204  007760  056767  171124  171064          BIS     WORK,DMA
1205  007766  026777  171060  171026          CMP     DMA,@RKDA       ;COMPARE LOWER 16 BITS
1206  007774  001423                          BEQ     CMDAE
1207  007776  004767  003424          ERADR:  JSR     %7,INCTAB
1208  010002  004567  002744                  JSR     %5,CONV         ;CONVERT DMA REG COUNT TO ASCII
1209  010006  001052                          DMA
1210  010010  013613                          MES1
1211  010012  000006                          6
1212  010014  104000                          EMT     +0              ;REPORT ERROR
1213  010016  013524                          MED4
1214  010020  013613                          MES1
1215  010022  177777                          -1
1216  010024  005067  171022                  CLR     DMA             ;DISK ADDRESS ERROR RE-START PROG.
1217  010030  032777  002000  170742          BIT     #BIT10,@SWR     ;HALT ON ERROR
1218  010036  001401                          BEQ     .+4
1219  010040  000000                          HALT                    ;SWITCH 10 SET IN SR HALT
1220  010042  000207                          RTS     %7
1221                                    ;
1222  010044  105767  170772          CMDAE:  TSTB    FLAG            ;CHECK FOR LAST DISK BUFFER
1223  010050  100013                          BPL     BUFINX
1224  010052  005067  170774          BUFEXIT: CLR    DMA             ;CLEAR ADDRESS BITS
1225  010056  062716  000002                  ADD     #2,(6)          ;INC STOCK POINTER
1226  010062  042767  000200  170752  AKH:    BIC     #200,FLAG
1227  010070  016767  170770  170750          MOV     SWRDCT,WRDCT
1228  010076  000515                          BR      EXTDR           ;EXIT
1229  010100  016767  170746  170742  BUFINX: MOV     DMA,TRACK       ;FETCH CYLINDER NUMBER
```

```
1230  010106  042767  160037  170734          BIC     #160037,TRACK   ;MASK CYLINDER NUMBER
1231  010114  005067  170774                  CLR     WORK2
1232  010120  016767  170726  170762          MOV     DMA,WORK
1233  010126  016767  170756  170762          MOV     WORK,WORK3
1234  010134  042767  177700  170746  XINCSEC: BIC    #177700,WORK    ;MASK SECTOR BITS
1235  010142  005267  170746                  INC     WORK2
1236  010146  022767  000013  170734          CMP     #13,WORK        ;CMP FOR LAST SECTOR
1237  010154  001406                          BEQ     XINCSUR         ;+1 SURFACE
1238  010156  005267  170726                  INC     WORK
1239  010162  005367  170724          CONTINC: DEC    WORK1
1240  010166  001362                          BNE     XINCSEC
1241  010170  000734                          BR      AKH
1242  010172  042767  000017  170710  XINCSUR: BIC    #17,WORK
1243  010200  032767  000020  170710          BIT     #BIT4,WORK3     ;TEST SURFACE
1244  010206  001004                          BNE     CLRSUR
1245  010210  052767  000020  170700          BIS     #BIT4,WORK3
1246  010216  000761                          BR      CONTINC
1247  010220  042767  000020  170670  CLRSUR: BIC     #BIT4,WORK3     ;CLEAR SURFACE BIT
1248  010226  062767  000040  170614          ADD     #40,TRACK       ;SET UP TEMP CYLINDER NUMBER
1249  010234  022767  014500  170606          CMP     #14500,TRACK    ;ARE WE AT THE LAST CYLINDER
1250  010242  001347                          BNE     CONTINC         ;BRANCH IF NO!
1251  010244  016767  170644  170574          MOV     WORK2,WRDCT
1252  010252  000241                          CLC
1253  010254  006167  170566                  ROL     WRDCT
1254  010260  006167  170562                  ROL     WRDCT
1255  010264  006167  170556                  ROL     WRDCT
1256  010270  006167  170552                  ROL     WRDCT
1257  010274  006167  170546                  ROL     WRDCT
1258  010300  006167  170542                  ROL     WRDCT
1259  010304  006167  170536                  ROL     WRDCT
1260  010310  032777  004000  170510          BIT     #BIT11,@RKDS    ;TEST FOR DENSITY
1261  010316  001402                          BEQ     .+6
1262  010320  006167  170522                  ROL     WRDCT
1263  010324  052767  000200  170510          BIS     #200,FLAG
1264  010332  000207                  EXTDR:  RTS     %7              ;EXIT
1265                                ;ROUTINE TO SET UP THE DISK NUMBER FOR THE ADDRESS TEST
1266                                ;
1267                                ;
1268                                ;
1269  010334  016767  170502  170546  DSKNOS: MOV     FLAG,WORK       ;FETCH THE FLAG WORD
1270  010342  006067  170542                  ROR     WORK
1271  010346  006067  170536                  ROR     WORK
1272  010352  000241                          CLC
1273  010354  006067  170530                  ROR     WORK
1274  010360  006067  170524                  ROR     WORK
1275  010364  006067  170520                  ROR     WORK
1276  010370  006067  170514                  ROR     WORK
1277  010374  042767  017777  170506          BIC     #17777,WORK     ;MASK THE DISK NUMBER
1278  010402  000205                          RTS     %5              ;EXIT
1279                                ;
1280                                ;
1281                                ;
```

```
1282
1283                                    ;
1284                                    ;
1285                                    ;
1286                                    ;
1287                                    ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
1288                                    ;
1289  010404  032777  004000  170414  BLGZ1:  BIT     #BIT11,@RKDS    ;CHECK DENSITY
1290  010412  001404                  BEQ     LODEN           ;DRIVE IS LOW DENSITY
1291  010414  012767  000377  170460  MOV     #377,BLOCK      ;DRIVE IS HI DENSITY
1292  010422  000403                  BR      SETBLK          ;SET UP NO. OF WORDS PER BLOCK
1293  010424  012767  000177  170450  LODEN:  MOV     #177,BLOCK      ;SET UP NO. WORDS PER BLOCK
1294  010432  016767  170410  170450  SETBLK: MOV     WRDCT,WORK      ;FETCH WORD COUNT
1295  010440  036767  170436  170442          BIT     BLOCK,WORK
1296  010446  001410                  BEQ     RORBLK
1297  010450  046767  170426  170432          BIC     BLOCK,WORK      ;SET UP BLOCK OVERFLOW
1298  010456  005267  170420          INC     BLOCK
1299  010462  066767  170414  170420          ADD     BLOCK,WORK
1300  010470  000241          RORBLK: CLC
1301  010472  006067  170412          ROR     WORK
1302  010476  006067  170406          ROR     WORK
1303  010502  006067  170402          ROR     WORK
1304  010506  006067  170376          ROR     WORK
1305  010512  006067  170372          ROR     WORK
1306  010516  006067  170366          ROR     WORK
1307  010522  006067  170362          ROR     WORK
1308  010526  032777  004000  170272          BIT     #BIT11,@RKDS    ;TEST FOR HI DENSITY
1309  010534  001402                  BEQ     .+6             ;LOW DENSITY
1310  010536  006067  170346          ROR     WORK
1311  010542  016767  170342  170332          MOV     WORK,BLOCK      ;BLOCK COUNT
1312  010550  000207                  RTS     %7              ;EXIT
1313                                    ;
1314                                    ;
1315                                    ;
1316                                    ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
1317                                    ;
1318                                    ;ENTER FROM JSR %5 PASEL
1319                                    ;
1320  010552  016700  170276          PASEL:  MOV     PATNU,%0        ;SET UP PATTERN NUMBER
1321  010556  010067  002250          MOV     %0,ACNVX
1322  010562  006067  002244          ROR     ACNVX
1323  010566  004567  002100          JSR     %5,CONV
1324  010572  013032                  ACNVX
1325  010574  014651                  PATMES
1326  010576  000002                  2
1327  010600  104000                  EMT+0
1328  010602  014655                  PATHED
1329  010604  014651                  PATMES
1330  010606  177777                  -1
1331  010610  016767  170232  170272          MOV     WRDCT,WORK      ;SET UP WORK
1332  010616  012701  014750          MOV     #OUTBUF,%1      ;LOC. OF OUTBUFFER
1333  010622  022700  000042          CMP     #42,%0          ;TEST FOR RANDOM DATA NUMBER
1334  010626  001406                  BEQ     RANDOM          ;GO GENERATE RANDOM DATA
1335  010630  016021  010776  FILDAT: MOV     PAT0(0),(1)+    ;FILL BUFFER
```

```
1336  010634  005367  170250              DEC     WORK            ;DEC. WORK COUNT
1337  010640  001373                      BNE     FILDAT          ;LOAD NEXT WORD
1338  010642  000205                      RTS     %5              ;BUFFER FULL
1339                                  ;
1340                                  ;RANDOM DATA
1341                                  ;
1342                                  ;
1343                                  ;
1344                                  ;
1345                                  ;RANDOM DATA GENERATOR SUBROUTINE
1346  010644  016700  000122  RANDOM: MOV     LONUM,%0        ;SET UP R0 WITH 5 DIGITS LOW
1347  010650  016704  000120          MOV     HINUM,%4        ;SET UP R1 WITH 5 DIGITS HIGH
1348  010654  012703  000007          MOV     #7,%3           ;SET UP SHIFT COUNT
1349  010660  005002                  CLR     %2              ;CLEAR R2
1350  010662  006300          SHIFT:  ASL     %0              ;SHIFT R0 LEFT AND
1351  010664  006104                  ROL     %4              ;ROTATE CARRY INTO LSB OF R1 INTO
1352  010666  006102                  ROL     %2              ;ROTATE CARRY OUT OF R1 INTO R2
1353  010670  005303                  DEC     %3              ;DECREMENT R3
1354  010672  001373                  BNE     SHIFT           ;CONTINUE SHIFT LOOP
1355  010674  066700  000072          ADD     LONUM,%0        ;ADDN IN NUMBER TO MAKE X 129
1356  010700  005504                  ADC     %4              ;PROPOGATE CARRY
1357  010702  066704  000066          ADD     HINUM,%4        ;ADDN IN NUMBER TO MAKE X 129
1358  010706  005502                  ADC     %2              ;PROPOGATE CARRY
1359  010710  062700  001057          ADD     #1057,%0        ;ADDN LOW CONSTANT
1360  010714  005504                  ADC     %4              ;PROPOGATE CARRIES
1361  010716  005502                  ADC     %2              ;PROPOGATE AGAIN
1362  010720  062704  047401          ADD     #47401,%4       ;ADDN HIGH CONSTANT
1363  010724  005502                  ADC     %2              ;PROPOGATE CARRY
1364  010726  062702  000006          ADD     #6,%2           ;ADDN HIGHEST CONSTANT
1365  010732  062700  000002          ADD     #2,%0           ;REPRIME R0 WITH HIGH DIGIT
1366  010736  005504                  ADC     %4              ;PROPOGATE CARRY
1367  010740  010067  000026          MOV     %0,LONUM        ;PUT R0 BACK IN LONUM
1368  010744  010021                  MOV     %0,(1)+         ;HOLD LONUM FOR PROGRAM
1369  010746  005367  170136          DEC     WORK
1370  010752  001406                  BEQ     EXGEN
1371  010754  010467  000014          MOV     %4,HINUM        ;PUT R1 BACK IN HINUM
1372  010760  010421                  MOV     %4,(1)+         ;HOLD HINUM FOR PROGRAM
1373  010762  005367  170122          DEC     WORK
1374  010766  001326                  BNE     RANDOM
1375  010770  000205          EXGEN:  RTS     %5              ;RETURN TO PROGRAM
1376  010772  000000          LONUM:  0
1377  010774  000000          HINUM:  0
1378                                  ;
1379                                  ;
1380                                  ;
1381                                  ;
1382                                  ;
1383                                  ;
1384                                  ;
```

```
1385
1386                                            .EVEN
1387                                            ;
1388                                            ;RK11 DATA PATTERNS
1389                                            ;
1390  010776  000000                            PAT0:   0
1391  011000  177777                            PAT1:   177777
1392  011002  134510                            PAT2:   134510
1393  011004  043267                            PAT3:   043267
1394  011006  100000                            PAT4:   100000
1395  011010  107070                            PAT5:   107070
1396  011012  070707                            PAT6:   070707
1397  011014  052525                            PAT7:   052525
1398  011016  125252                            PAT10:  125252
1399  011020  177737                            PAT11:  177737
1400  011022  004102                            PAT12:  004102
1401  011024  136363                            PAT13:  136363
1402  011026  063636                            PAT14:  063636
1403  011030  000001                            PAT15:  000001
1404  011032  100005                            PAT16:  100005
1405  011034  000520                            PAT17:  000520
1406  011036  030303                            PAT20:  030303
1407                                            ;PAT21  RANDOM DATA
1408                                            ;
1409                                            ;
1410                                            ;
1411                                            ;DATA COMPARISON ROUTINE
1412                                            ;IF AN ERROR OCCURS BETWEEN THE OUT-BUFFER AND
1413                                            ;THE IN-BUFFER AN ERROR WILL BE REPORTED IN THE
1414                                            ;
1415                                            ;
1416  011040  012767  177766  170020  COMPARE:  MOV     #-10.,ERCOUNT   ;REPORT ONLY 10 ERRORS PER BUFFER
1417  011046  005067  170042                   CLR     WORK2           ;WORD COUNT
1418  011052  005067  170034                   CLR     WORK1
1419  011056  016767  167770  170024           MOV     DMA,WORK
1420  011064  032777  004000  167734           BIT     #BIT11,@RKDS
1421  011072  001404                           BEQ     BLNK
1422  011074  012767  000400  170000           MOV     #400,BLOCK      ;HI DENSITY DRIVE
1423  011102  000403                           BR      .+10
1424  011104  012767  000200  167770  BLNK:   MOV     #200,BLOCK      ;LO DENSITY DRIVE
1425  011112  012767  014750  167750           MOV     #OUTBUF,SAVE    ;SET UP OUTBUFFER POINTER
1426  011120  016767  167762  167744           MOV     INBUF,SAV1      ;SET UP IN BUFFER POINTER
1427  011126  027777  167736  167736  WRDCMP: CMP     @SAVE,@SAV1     ;COMPARE BUFFERS
1428  011134  001051                           BNE     WDERR           ;WORD IN ERROR
1429  011136  005267  167752           WRDINC: INC     WORK2           ;+1 WORD COUNT
1430  011142  005267  167744                   INC     WORK1
1431  011146  026767  167674  167740           CMP     WRDCT,WORK2     ;IS COMPLETE BUFFER CHECKED
1432  011154  001440                           BEQ     ADAM            ;EXIT ROUTINE
1433  011156  036767  167720  167726           BIT     BLOCK,WORK1
1434  011164  001425                           BEQ     BLAD1
1435  011166  016767  167716  167716           MOV     WORK,WORK1
1436  011174  042767  177700  167710           BIC     #177700,WORK1
1437  011202  022767  000013  167702           CMP     #13,WORK1
1438  011210  001007                           BNE     BALD1X
```

```
1439  011212  052767  000020  167670          BIS     #BIT4,WORK
1440  011220  042767  000017  167662          BIC     #17,WORK
1441  011226  000402                          BR      .+6
1442  011230  005267  167654          BALD1X: INC     WORK
1443  011234  005067  167652                  CLR     WORK1
1444  011240  062767  000002  167622  BLAD11: ADD     #2,SAVE
1445  011246  062767  000002  167616          ADD     #2,SAV1
1446  011254  000724                          BR      WRDCMP          ;COMPARE NEXT WORD
1447  011256  000205                  ADAM:   RTS     %5              ;EXIT THIS ROUTINE
1448                                          ;
```

```
1449
1450  011260  004767  002142          WDERR1  JSR     X7,INCTAB
1451  011264  052767  001000  167550          BIS     #BIT9,FLAG      ;SET ERROR BIT
1452  011272  005267  167570                  INC     ERCOUNT         ;+1 ERROR COUNT
1453  011276  005767  167564                  TST     ERCOUNT         ;WHEN COUNT = 0 EXIT
1454  011302  001765                          BEQ     ADAM            ;COUNT ZERO EXIT
1455  011304  004567  001442                  JSR     X5,CONV         ;CONVERT WORD ADDR TO ASCII
1456  011310  001110                          WORK
1457  011312  013613                          MES1
1458  011314  000006                          6
1459  011316  004567  001430                  JSR     X5,CONV
1460  011322  001112                          WORK1
1461  011324  013721                          MES4
1462  011326  000004                          4
1463  011330  017767  167534  001474          MOV     @SAVE,ACNVX     ;FETCH GOOD DATA
1464  011336  004567  001410                  JSR     X5,CONV         ;CONVERT GOOD DATA TO ASCII
1465  011342  013032                          ACNVX
1466  011344  013737                          MES5
1467  011346  000006                          6
1468  011350  017767  167516  001454          MOV     @SAV1,ACNVX     ;FETCH BAD DATA
1469  011356  004567  001370                  JSR     X5,CONV         ;CONVERT TO ASCII
1470  011362  013032                          ACNVX
1471  011364  013737                          MES6
1472  011366  000006                          6
1473  011370  016767  167446  001434          MOV     FLAG,ACNVX      ;WHICH READ THE
1474  011376  042767  177774  001426          BIC     #177774,ACNVX   ;ERROR OCCURRED ON
1475  011404  004567  001342                  JSR     X5,CONV
1476  011410  013032                          ACNVX
1477  011412  014020                          MES13
1478  011414  000001                          1
1479  011416  104000                          EMT     +0              ;PRINT MESSAGE
1480  011420  013470                          MED1
1481  011422  014020                          MES13
1482  011424  013613                          MES1
1483  011426  013721                          MES4
1484  011430  013737                          MES5
1485  011432  013757                          MES6
1486  011434  177777                          -1
1487  011436  032777  002000  167334          BIT     #BIT10,@SWR     ;HALT ON ERROR
1488  011444  001401                          BEQ     .+4
1489  011446  000000                          HALT                    ;YES HALT BIT 10 SET IN SR
1490  011450  000632                          BR      WRDINC          ;GO COMPARE NEXT WORD
1491
1492
1493
1494
1495
1496
```

```
1497
1498                                    ;
1499                                    ;
1500                                    ;
1501                                    ;
1502                                    ;
1503                                    ;
1504                                    ;
```

```
1505
1506                                    ;THE PROGRAM DETERMINES HOW MUCH MEMORY
1507                                    ;IS ON THE SYSTEM
1508                                    ;
1509  011452  012777  000001  167334  EXPMEN: MOV     #BIT0,@RKCS     ;CLEAR THE DISK
1510  011460  105777  167330          TSTB    @RKCS
1511  011464  100375                  BPL     .-4
1512  011466  012777  000340  167306          MOV     #340,@PS        ;LOCK UP PRIORITY LEVELS
1513  011474  012767  011552  166302          MOV     #MAXREF,4       ;SET UP I/O BUS TRAP
1514  011502  012767  000340  166276          MOV     #340,6
1515  011510  005067  167354          CLR     SAVE            ;SET UP FOR 4K
1516  011514  005777  167350  EXREF:  TST     @SAVE           ;REFERENCE MEMORY
1517  011520  022767  160000  167342          CMP     #160000,SAVE    ;TEST FOR GREATER THAN 28K
1518  011526  001001                  BNE     .+4             ;BRANCH IF LESS THAN 28K
1519  011530  000410                  BR      MAXREF          ;LAST REFERENCE MADE TO I/O REG.
1520  011532  062767  020000  167330          ADD     #20000,SAVE     ;SET UP FOR NEXT MEMORY REF.
1521  011540  000765                  BR      EXREF           ;GO REFERENCE MEMORY
1522  011542  162767  000020  167340  LGMEM:  SUB     #20,WORK        ;DEC IT
1523  011550  000424                  BR      MYWAY-6
1524                                    ;
1525                                    ;ENTER HERE WHEN I/O BUS ERROR OCCURS
1526                                    ;
1527  011552  012767  000006  166224  MAXREF: MOV     #6,4            ;RESTORE I/O BUS TRAP
1528  011560  005067  166222          CLR     6
1529  011564  162767  000500  167276          SUB     #500, SAVE
1530  011572  013767  000042  167310          MOV     @#42, WORK
1531  011600  001413                  BEQ     MYWAY
1532  011602  100757                  BMI     LGMEM           ;GREATER THAN 16K
1533  011604  162767  000020  167276          SUB     #20, WORK
1534  011612  022767  014750  167270          CMP     #OUTBUF, WORK
1535  011620  003003                  BGT     MYWAY
1536  011622  016767  167262  167240          MOV     WORK, SAVE
1537  011630  162767  014750  167232  MYWAY:  SUB     #OUTBUF,SAVE    ;SET UP NEW WORD COUNT
1538  011636  006067  167226          ROR     SAVE
1539  011642  016767  167222  167176          MOV     SAVE,WRDCT      ;
1540  011650  042767  000001  167212          BIC     #BIT0,SAVE
1541  011656  012767  014750  167222          MOV     #OUTBUF,INBUF
1542  011664  066767  167200  167214          ADD     SAVE,INBUF
1543  011672  006067  167150          ROR     WRDCT
1544  011676  042767  000377  167142          BIC     #377,WRDCT
1545  011704  016767  167136  167152          MOV     WRDCT,$WRDCT
1546  011712  000205                  RTS     %5
1547                                    ;BACKGROUND TEST FOR INTERRUPTS
1548                                    ;
1549  011714  012767  011730  166066  XWAIT:  MOV     #MODRTX,10
1550  011722  006727  000000          SXT     #0
1551  011726  000404                  BR      SETPAS
1552  011730  012767  000002  000102  MODRTX: MOV     #2,RTIX
1553  011736  022626                  CMP     (6)+,(6)+
1554  011740  012767  012040  166046  SETPAS: MOV     #RTIX,14
1555  011746  012767  000012  166034          MOV     #12,10
1556  011754  012746  000020          MOV     #BIT4,-(6)
1557  011760  012746  011766          MOV     #.+6,-(6)
1558  011764  000002                  RTI
```

```
1559  011766  005067  167102                CLR     PASS            ;SET UP TIME BASE
1560  011772  005027  000000                CLR     #0
1561  011776  005267  177772        XINCW:  INC     XINCW-2
1562  012002  105767  177766                TSTB    XINCW-2
1563  012006  100373                        BPL     XINCW
1564  012010  005267  167060                INC     PASS
1565  012014  001401                        BEQ     .+4
1566  012016  000765                        BR      XINCW-4
1567                                ;REPORT BACKGROUND TEST TIMED OUT
1568  012020  012746  000340                MOV     #340,@(6)       ;CLEAR TRACE BIT
1569  012024  012746  012032                MOV     #.+6,@(6)
1570  012030  000002                        RTI
1571  012032  104001                        EMT+1
1572  012034  014030                        TIMO
1573  012036  000000                        HALT
1574                                ;
1575  012040  000006                RTIX:   RTT
1576
```

```
                                        ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
                                        ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
                                        ;
                                        ;WORD COUNT REGISTER
012042  017777  166732  166746  SELWC:  MOV     @SWR,@RKWC              ;MOV SR INTO WORD COUNT REG
012050  000774                          BR      SELWC
                                        ;
                                        ;CURRENT ADDRESS REGISTER
012052  017777  166722  166740  SELCMA: MOV     @SWR,@RKBA              ;MOV SR INTO CURRENT ADDR REG
012060  000774                          BR      SELCMA
                                        ;
                                        ;DISK ADDRESS REGISTER
012062  017777  166712  166732  SELDAR: MOV     @SWR,@RKDA              ;MOV SR INTO DISK ADDR REG
012070  000774                          BR      SELDAR
                                        ;
                                        ;
                                        ;
                                        ;DISK CONTROL STATUS REGISTER
012072  012777  000340  166702  SELDCS: MOV     #340,@PS                ;LOCK UP INTERRUPTS
012100  012777  177777  166710          MOV     #177777,@RKWC           ;SET WORD COUNT =1 WORD
012106  012777  014750  166704          MOV     #OUTBUF,@RKBA           ;SET UP CURRENT ADDRESS
012114  017777  166660  166672          MOV     @SWR,@RKCS              ;MOV SR INTO CONTROL REG
012122  032777  000001  166650          BIT     #BIT0,@SWR              ;IS FUNCTION BITS SET
012130  001760                          BEQ     SELDCS                  ;FUNCTION BITS NOT SET
012132  105777  166656          DKBUSY: TSTB    @RKCS                   ;TEST FOR DISK NOT READY
012136  100375                          BPL     DKBUSY                  ;DISK STILL READY
012140  000754                          BR      SELDCS                  ;DISK NOT BUSY SELECT NEW CR
                                        ;
                                        ;
                                        ;POWER DOWN ROUTINE
                                        ;WAIT FOR THE TRANSFER TO BE COMPLETE
                                        ;
                                        ;
012142  012767  012166  165654  DOWN:   MOV     #UP,24                  ;SET UP POWER FAIL VECTOR
012150  012777  000001  166636          MOV     #BIT0,@RKCS
012156  105777  166632                  TSTB    @RKCS
012162  100375                          BPL     .-4
012164  000000                          HALT
                                        ;
                                        ;
                                        ;
012166  012767  012142  165630  UP:     MOV     #DOWN,24                ;SET UP FOR NEXT POWER FAIL
012174  104001                          EMT+1                           ;REPORT PWR FAIL
012176  014103                          PWRF
012200  105777  166622                  TSTB    @RKDS                   ;TEST FOR DRIVE READY
012204  100375                          BPL     .-4
012206  000167  166706                  JMP     START                   ;GO CHECK DRIVE
                                        ;
                                        ;
                                        ;
                                        ;THIS ROUTINE ENABLES THE OPERATOR TO SELECT A TRACK STATICLY
                                        ;THE ROUTINE DOES A ONE WORD READ TO SELECT THE TRACK
                                        ;THE OPERATOR MAY CHANGE THE SWITCH REGISTER AT ANY TIME
```

```
1631                                        ;SET DRIVE AND CYLINDER NO. IN SR
1632                                        ;
1633  012212  012777  000001  166574  TKSEL1: MOV     #BIT0,@RKCS
1634  012220  105777  166570                 TSTB    @RKCS           ;TEST FOR READY
1635  012224  100375                         BPL     .-4
1636  012226  017767  166546  166656         MOV     @SWR,WORK1      ;FETCH SR
1637  012234  016767  166652  166646         MOV     WORK1,WORK
1638  012242  042767  174000  166640         BIC     #174000,WORK
1639  012250  000241                         CLC
1640  012252  006167  166632                 ROL     WORK
1641  012256  006167  166626                 ROL     WORK
1642  012262  006167  166622                 ROL     WORK
1643  012266  006167  166616                 ROL     WORK
1644  012272  006167  166612                 ROL     WORK
1645  012276  016777  166606  166516         MOV     WORK,@RKDA      ;DISK ADDRESS LOADED
1646  012304  016777  166576  166506         MOV     INBUF,@RKBA     ;LOAD CURRENT ADDRESS
1647  012312  012777  177777  166476         MOV     #177777,@RKWC   ;LOAD WORD COUNT
1648  012320  012777  000011  166466         MOV     #11,@RKCS       ;GO AND READ
1649  012326  105777  166462                 TSTB    @RKCS           ;TEST FOR READY
1650  012332  100375                         BPL     .-4
1651  012334  032777  000100  166464  CTBUSY: BIT     #BIT6,@RKDS
1652  012342  001774                         BEQ     CTBUSY          ;WAIT FOR CONTROL NOT READY
1653  012344  026777  166542  166426  SRCHG:  CMP     WORK1,@SWR
1654  012352  001317                         BNE     TKSEL           ;SWR HAS CHANGED
1655  012354  000773                         BR      SRCHG           ;SWR HAS NOT CHANGED
1656                                        ;
1657                                        ;
1658                                        ;
1659                                        ;
1660                                        ;
1661                                        ;
```

```
1662
1663                                    ;
1664                                    ;ROUTINE TO REPORT ERROR COUNT AND CONTENTS OF ONE REGISTER
1665                                    ;
1666  012356  004767  001044    STAER1  JSR     X7,INCTAB
1667  012362  004567  000364            JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1668  012366  001110                    WORK                    ;DATA TO BE CONVERTED
1669  012370  013757                    MES6                    ;ADDRESS OF MESSAGE
1670  012372  000006                    6
1671  012374  004567  000352            JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1672  012400  001066                    ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
1673  012402  013554                    HED5                    ;ADDRESS OF MESSAGE
1674  012404  000003                    3
1675  012406  104000                    EMT     +0              ;REPORT MESSAGE
1676  012410  013552                    HED5A
1677  012412  013554                    HED5
1678  012414  013757                    MES6
1679  012416  177777                    -1
1680  012420  000205                    RTS     X5              ;EXIT ROUTINE
1681                                    ;
1682                                    ;ROUTINE TO REPORT ERROR COUNT AND THE CONTENTS OF TWO REGISTERS
1683                                    ;
1684                                    ;
1685                                    ;
1686  012422  004767  001000    STAER11 JSR     X7,INCTAB
1687  012426  004567  000320            JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1688  012432  001110                    WORK                    ;DATA TO BE CONVERTED
1689  012434  013757                    MES6                    ;ADDRESS OF MESSAGE
1690  012436  000006                    6
1691  012440  004567  000306            JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1692  012444  001112                    WORK1                   ;DATA TO BE CONVERT
1693  012446  013737                    MES5                    ;ADDRESS OF MESSAGE
1694  012450  000006                    6
1695  012452  004567  000274            JSR     X5,CONV         ;CONVERT OCTAL TO ASCII
1696  012456  001066                    ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
1697  012460  013554                    HED5                    ;ADDRESS OF MESSAGE
1698  012462  000003                    3
1699  012464  104000                    EMT     +0              ;REPORT MESSAGE
1700  012466  013552                    HED5A
1701  012470  013554                    HED5
1702  012472  013757                    MES6
1703  012474  013737                    MES5
1704  012476  177777                    -1
1705  012500  000205                    RTS     X5              ;EXIT ROUTINE
1706                                    ;
1707                                    ;
1708                                    ;ROUTINE TO DECODE EMT CALLS
1709                                    ;EMT+1=TYPE ONE LINE OF TEXT
1710                                    ;EMT+0=TYPE A SERIES OF LINES
1711  012502  011600            EMTRP1  MOV     (0),X0
1712  012504  022740  104001            CMP     #EMT+1,-(0)     ;WAS THE CALL EMT+1
1713  012510  001103                    BNE     TYPS            ;NO! TYPE A SERIES OF LINES OF TEXT
1714  012512  000400                    BR      TYP             ;YES TYPE ONE LINE OF TEXT
```

```
1715
1716                                    ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER,
1717  012514  011600                    TYP:    MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1718  012516  062716  000002                    ADD     #2,@%6          ;SET UP EXIT,
1719  012522  011000                            MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0,
1720  012524  112067  000164            TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
1721  012530  122767  000100  000156            CMPB    #100,TYPDAT     ;CHECK FOR "@" CHARACTER
1722  012536  001005                            BNE     TYPC            ;BRANCH IF NOT "@",
1723  012540  005067  000150                    CLR     TYPDAT          ;OUTPUT NULL
1724  012544  004767  000030                    JSR     %7,TYPD         ;TO CLEAR DOUBLE BUFFER
1725  012550  000002                            RTI                     ;TERMINATOR CHAR, DONE, EXIT,
1726  012552  122767  000045  000134  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR "%",
1727  012560  001442                            BEQ     TYPF            ;BRANCH IF "%",
1728  012562  122767  000042  000124            CMPB    #42,TYPDAT      ;NOT "%", CHECK FOR "#",
1729  012570  001443                            BEQ     TYPG            ;BRANCH IF "#"
1730  012572  004767  000002                    JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
1731  012576  000752                            BR      TYPA
1732  012600  032777  040000  166172  TYPD:   BIT     #BIT14,@SWR
1733  012606  001026                            BNE     TYEXIT
1734  012610  116777  000100  166166            MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
1735  012616  105777  166166                    TSTB    @TPS            ;WAIT FOR DONE FLAG,
1736  012622  100375                            BPL     .-4
1737  012624  122767  000015  000062            CMPB    #15, TYPDAT     ;CHECK FOR CR
1738  012632  001003                            BNE     1$              ;NO - SKIP
1739  012634  012767  000011  000054            MOV     #9., NULL       ;SET NULL COUNTER
1740  012642  005767  000050            1$:     TST     NULL            ;TEST COUNTER
1741  012646  001406                            BEQ     TYEXIT          ;ZERO-EXIT
1742  012650  005367  000042                    DEC     NULL            ;DEC COUNTER
1743  012654  112767  000000  000032            MOVB    #0,TYPDAT       ;ZERO OUTPUT
1744  012662  000746                            BR      TYPD            ;OUTPUT NULL
1745  012664  000207                    TYEXIT: RTS     %7              ;EXIT
1746  012666  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1747  012674  004767  177700                    JSR     %7,TYPD         ;GO TYPE CHAR,
1748  012700  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT,
1749  012706  004767  177666                    JSR     %7,TYPD         ;GO TYPE CHAR,
1750  012712  000704                            BR      TYPA
1751  012714  000000                    TYPDAT: 0
1752  012716  000000                    NULL:   0
1753                                    ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
1754  012720  011600                    TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1755  012722  062716  000002                    ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
1756  012726  011067  000014                    MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
1757  012732  022767  177777  000006            CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
1758  012740  001001                            BNE     TYPSA           ;BRANCH IF NOT TERMINATOR,
1759  012742  000002                            RTI                     ;TERMINATOR, EXIT
1760  012744  104001                    TYPSA:  EMT     +1              ;CALL ON TYP SUB TO TYPE MESSAGE
1761  012746  000000                    TYPSB:  0                       ;ADDRESS OF MESSAGE GOES HERE
1762  012750  000763                            BR      TYPS            ;GO PROCESS NEXT MESSAGE
1763                                    ;
1764                                    ;
1765                                    ;
1766                                    ;
1767                                    ;OCTAL TO ASCII CONVERT ROUTINE
1768                                    ;
```

```
1769                                    ;ENTER ROUTINE AS FOLLOWS
1770                                    ;JSR%5,CONV
1771                                    ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED
1772                                    ;ADDR BYTE=LSB OF WHERE ASCII IS GOING
1773                                    ;ASCII#=THE NUMBER OF ASCII CHAR, TO BE CONVERTED
1774                                    ;
1775                                    ;
1776  012752  013567  000054           CONV:   MOV     @(5)+,ACNVX     ;VALUE OF # TO BE CONVERTED
1777  012756  012501                            MOV     (5)+,%1         ;ASCII ADDR
1778  012760  012502                            MOV     (5)+,%2         ;# OF ASCII CHAR
1779  012762  060201                            ADD     %2,%1
```

```
1780  012764  016703  000042            ACVNI:  MOV     ACNVX,X3
1781  012770  042703  177770                    BIC     #177770,X3      ;ISOLATE LEAST SIGNIFICANT OCTAL#
1782  012774  062703  000060                    ADD     #60,X3          ;SET UP ASCII#
1783  013000  110341                           MOVB    X3,-(1)         ;STORE ASCII CHAR
1784  013002  042767  000007  000022           BIC     #7,ACNVX
1785  013010  006067  000016                    ROR     ACNVX           ;ROTATE OCTAL#
1786  013014  006067  000012                    ROR     ACNVX
1787  013020  006067  000006                    ROR     ACNVX
1788  013024  005302                           DEC     X2              ;-1 FROM ASCII CHAR COUNT
1789  013026  001356                           BNE     ACVN
1790  013030  000205                           RTS     X5              ;EXIT # CONVERTED
1791  013032  000000            ACNVX:  0                               ;WORK REGISTER
1792                            ;
1793                            ;
1794                            ;
1795                            ;
1796                            ;ROUTINE TO HANDLE KEYBOARD CONVERSATION
1797                            ;ROUTINE ACCEPTS NUM, OR ALPHA CHARACTERS
1798                            ;
1799  013034  012767  000040  000362    NOCHA:  MOV     #40,TSTCH       ;SET UP FOR NUM, CHAR,
1800  013042  000403                           BR      TYST
1801  013044  012767  000100  000352    ALPHA:  MOV     #100,TSTCH      ;SET UP FOR ALPHA CHAR
1802  013052  012777  000340  165722    TYST:   MOV     #340,@PS        ;LOCK UP INTERRUPTS
1803  013060  005067  000160                    CLR     TEXBUF          ;CLEAR TEXT BUFFER REG
1804  013064  105777  165722            TSTFLG: TSTB    @TKS            ;CHECK FOR FLAG
1805  013070  100375                           BPL     TSTFLG          ;
1806  013072  017777  165710  165704            MOV     @TKB,@TPB       ;CHARACTER IN BUFFER
1807  013100  105777  165704                    TSTB    @TPS            ;ECHO CHARACTER
1808  013104  100375                           BPL     .-4
1809  013106  022777  000377  165672            CMP     #377,@TKB       ;CHECK FOR RUB-OUT
1810  013114  001014                           BNE     CKCH            ;EXIT IF NOT RUB-OUT
1811  013116  104001                           EMT     +1
1812  013120  014001                           MES8                    ;REPORT RUB-OUT ACKNOWLEDGED
1813  013122  042767  000007  000114           BIC     #7,TEXBUF
1814  013130  006067  000110                    ROR     TEXBUF
1815  013134  006067  000104                    ROR     TEXBUF
1816  013140  006067  000100                    ROR     TEXBUF
1817  013144  000747                           BR      TSTFLG          ;GO WAIT FOR NEW CHAR,
1818  013146  022777  000215  165632    CKCH:   CMP     #215,@TKB       ;CHECK FOR CARRIAGE RETURN
1819  013154  001001                           BNE     .+4
1820  013156  000207                           RTS     X7              ;EXIT DELIMITER TYPED
1821  013160  036777  000240  165620            BIT     TSTCH,@TKB
1822  013166  001003                           BNE     CHOK
1823  013170  104001                           EMT     +1              ;REPORT QUESTION MARK
1824  013172  013777                           MES7
1825  013174  000733                           BR      TSTFLG          ;WAIT FOR CORRECT CHAR,
1826  013176  017767  165604  165704    CHOK:   MOV     @TKB,WORK
1827  013204  042767  177770  165676            BIC     #177770,WORK
1828  013212  000241                           CLC
1829  013214  006167  000024                    ROL     TEXBUF
1830  013220  000241                           CLC
1831  013222  006167  000016                    ROL     TEXBUF
1832  013226  000241                           CLC
1833  013230  006167  000010                    ROL     TEXBUF
```

```
1834  013234  066767  163650  000002          ADD     WORK,TEXBUF     ;ADD CHARACTER
1835  013242  000710                          BR      TSTFLG          ;WAIT FOR NEW CHARACTER
```

```
1836  013244  000000                    TEXBUF: 0
1837                                    ;ROUTINE TO SET ACTION ENABLE ON MA/MF PARITY MEMORIES
1838                                    ;CALL   JSR PC,MAMF
1839          172100                            PARCSR=172100
1840          000114                            PARVEC=114
1841          000004                            ERRVEC=4
1842          000006                                SP=%6
1843  013246  012737  013340  000114  MAMF:   MOV     # PARSRV, @#PARVEC       ;SET PARITY INTERRUPT VECTOR
1844  013254  012737  000340  000116          MOV     #340, @#PARVEC+2        ;AND PRIORITY LEVEL 7 ON INTERRUPT
1845  013262  013746  000004                  MOV     @#ERRVEC,-(SP)          ;SAVE CURRENT ERROR VECTOR
1846  013266  013746  000006                  MOV     @#ERRVEC+2,-(SP)        ;AND PRIORITY LEVEL
1847  013272  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC
1848  013300  012737  000002  000006          MOV     #RTI,@#ERRVEC+2
1849  013306  012700  172100                  MOV     #PARCSR,%0              ;GET FIRST CSR ADDRESS
1850  013312  012702  000001                  MOV     #1,%2
1851  013316  012720  000001          1$:     MOV     #1,(0)+                 ;SET ACTION ENABLE IF AVAILABLE
1852  013322  006302                          ASL     %2                      ;SHIFT AVAILABILITY INDICATOR
1853  013324  103374                          BCC     1$
1854  013326  012637  000006                  MOV     (SP)+,@#ERRVEC+2        ;RESTORE ERROR VECTOR
1855  013332  012637  000004                  MOV     (SP)+,@#ERRVEC          ;PRIORITY LEVEL AND INTERRUPT VECTOR
1856  013336  000207                          RTS     %7
1857                                    ;PARITY ERROR SERVICE ROUTINE
1858                                    ;WHEN A PARITY ERROR IS DETECTED THE ROUTINE SCANS
1859                                    ;MEMORY FOR THE PARITY ERROR A WHEN THE ERROR
1860                                    ;IS DETECTED THE PROGRAM HALTS WITH THE ADDRESS
1861                                    ;CAUSING THE ERROR IN R0
1862                                    ;TO CONTINUE PRESS CONTINUE
1863  013340  104001          PARSRV: EMT+1
1864  013342  014056                  PARERR
1865  013344  012737  013372  000114          MOV     #2$,@#PARVEC    ;REPOSITION PARITY ERROR INT.
1866  013352  012737  013420  000004          MOV     #4$,@#ERRVEC            ;SET TIME OUT TRAP
1867  013360  005037  000006                  CLR     @#ERRVEC+2
1868  013364  005000                          CLR     %0
1869  013366  005720          1$:     TST     (0)+                    ;SCAN MEMORY
1870  013370  000776                  BR      1$
1871  013372  000000          2$:     HALT                            ;PARITY ERROR=ADDRESS
1872                                                                  ;CAUSING ERROR IS IN REGISTER0
1873  013374  000005          3$:     RESET
1874  013376  012737  013340  000114          MOV     #PARSRV,@#PARVEC        ;RESTORE PARITY VECTOR
1875  013404  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;RESTORE TIMEOUT HALT
1876  013412  004767  177630                  JSR     %7,MAMF
1877  013416  000002                          RTI
1878  013420  000000          4$:     HALT                            ;ERROR=PARITY ERROR NOT DECTECTED ON SCAN
1879  013422  000764                  BR      3$                      ;4(SP) CONTAINS PC WHERE
1880                                                                  ;PARITY ERROR WAS ORIGINALLY DETECTED
1881  013424  000000          TSTCH:  0
1882                                    ;
1883                                    ;
1884  013426  016703  165410  INCTAB: MOV     FLAG,%3
1885  013432  042703  177743          BIC     #177743,%3
1886  013436  000241                  CLC
1887  013440  006003                  ROR     %3
1888  013442  005263  013450          INC     TABLE(3)
1889  013446  000207                  RTS     %7
```

```
                                TABLE1  0                       ;ERROR COUNT DRIVE 0
                                        0                       ;ERROR COUNT DRIVE 1
                                        0                       ;ERROR COUNT DRIVE 2
                                        0                       ;ERROR COUNT DRIVE 3
                                        0                       ;ERROR COUNT DRIVE 4
                                        0                       ;ERROR COUNT DRIVE 5
                                        0                       ;ERROR COUNT DRIVE 6
                                        0                       ;ERROR COUNT DRIVE 7
                                ;
                                ;
                                ;
                                ;ERROR MESSAGE HEADERS
                                ;
                                ;
                                        .EVEN
013470  042045  052101  020101  HED1:   .ASCII  /%DATA ERROR @/
013476  051105  047522  020122
013504     100
                                ;
                                ;
013505     045  052123  052101  HED2:   .ASCII  /%STATUS ERROR @/
013512  051525  042440  051122
013520  051117  040040
                                ;
                                ;
013524  042045  044522  042526  HED4:   .ASCII  /%DRIVE ADDRESS ERROR @/
013532  040440  042104  042522
013540  051523  042440  051122
013546  051117  040040
                                ;
                                ;
013552  040045                  HED5A:  .ASCII  /%@/
                                ;
                                ;
013554  020040  020040  051105  HED5:   .ASCII  /    ERROR COUNT @/
013562  047522  020122  047503
013570  047125  020124     100
                                ;
                                ;
013575     045  040522  042516  HED6:   .ASCII  /%RANEX ERROR @/
013602  020130  051105  047522
013610  020122     100
                                ;
                                ;
                                ;MESSAGE TRAILERS
                                ;
                                ;
013613     040  020040  020040  MEG1:   .ASCII  /       RKDA @/
013620  020040  045522  040504
013626  040040
                                ;
                                ;
                                ;
013630  020040  020040  020040  MEG1A:  .ASCII  /       RKER @/
```

```
1944  013636  051040  042513  020122
1945  013644     100
1946                              ;
1947                              ;
1948                              ;
1949  013645     040  020040  020040  MESG2:  .ASCII  /      RKCS @/
1950  013652  020040  045522  051503
1951  013660  040040
1952                              ;
1953  013662  020040  020040  020040  MESG2A:  .ASCII  /      RKDS @/
1954  013670  051040  042113  020123
1955  013676     100
1956                              ;
1957  013677     040  020040  020040  MESG3:  .ASCII  /      WORD COUNT@/
1958  013704  020040  047527  042122
1959  013712  041440  052517  052116
1960  013720     100
1961                              ;
1962                              ;
1963                              ;
1964                              ;
1965  013721     040  020040  020040  MESG4:  .ASCII  /     WRD ADR.@/
1966  013726  051127  020104  042101
1967  013734  027122     100
1968                              ;
1969                              ;
1970                              ;
1971  013737     040  020040  020040  MESG5:  .ASCII  /      GD DATA @/
1972  013744  020040  042107  042040
1973  013752  052101  020101     100
1974                              ;
1975                              ;
1976                              ;
```

```
1977
1978  013757     040  020040  020040  MEG6:  .ASCII  /       BD DATA @/
1979  013764  020040  042102  042040
1980  013772  052101  020101     100
1981                                   )
1982                                   )
1983                                   )
1984  013777     077     100          MEG7:  .ASCII  /?@/
1985                                   )
1986                                   )
1987                                   )
1988                                   )
1989  014001     057     100          MEG8:  .ASCII  '/@'
1990                                   )
1991                                   )
1992                                   )
1993                                   )
1994  014003     045  047125  052111  MEG11: .ASCII  /%UNIT NO:@/
1995  014010  047040  027117     100
1996                                   )
1997                                   )
1998  014015     040  040040          MEG12: .ASCII  /  @/
1999                                   )
2000                                   )
2001  014020  020040  042522  042101  MEG13: .ASCII  /  READ @/
2002  014026  040040
2003                                   )
2004                                   )
2005  014030  041445  052520  041040  TIMO:  .ASCII  /%CPU BKGRND TIMED OUT@/
2006  014036  043513  047122  020104
2007  014044  044524  042515  020104
2008  014052  052517  040124
2009                                   )
2010                                   )
2011                                   )
2012  014056  046445  046505  051117  PARERR: .ASCII  /%MEMORY PARITY ERROR@/
2013  014064  020131  040520  044522
2014  014072  054524  042440  051122
2015  014100  051117     100
2016                                   )
2017                                   )
2018  014103     045  047520  042527  PWRF:  .ASCII  /%POWER HAS FAILED@/
2019  014110  020122  040510  020123
2020  014116  040506  046111  042105
2021  014124     100
```

```
2022
2023                                        ;CONVERSATION TEXT
2024                                        ;
2025                                        ;
2026  014125     040  020040  020040       SPECMES:  .ASCII  /       SWRDCT?@/
2027  014132  020040  053523  042122
2028  014140  052103     100
2029                                        ;;
2030  014143     045  040504  040524       CON1:   .ASCII  /%DATA TEST ONLY? (YES=NO)?@/
2031  014150  052040  051505  020124
2032  014156  047117  054514  020077
2033  014164  054450  051505  047055
2034  014172  024517     100
2035                                        ;
2036                                        ;
2037                                        ;
2038  014175     045  052515  052114       CON2:   .ASCII  /%MULTI DRIVE MODE?(YES=NO)?@/
2039  014202  020111  051104  053111
2040  014210  020105  047515  042504
2041  014216  024077  042531  026523
2042  014224  047516  040051
2043                                        ;
2044                                        ;
2045  014230  047045  046525  042502       CON3:   .ASCII  /%NUMBER OF DRIVES 1 TO 10 OCTAL?@/
2046  014236  020122  043117  042040
2047  014244  044522  042526  020123
2048  014252  020061  047524  030440
2049  014260  020060  041517  040524
2050  014266  037514     100
2051                                        ;
2052                                        ;
2053  014271     045  054105  051105       CON4:   .ASCII  /%EXERCISE DRIVE?@/
2054  014276  044503  042523  042040
2055  014304  044522  042526  040077
2056                                        ;
2057                                        ;
2058  014312  047445  052120  047511       CON5:   .ASCII  /%OPTIONAL WORD COUNT (YES=NO)?@/
2059  014320  040516  020114  047527
2060  014326  042122  041440  052517
2061  014334  052116  024040  042531
2062  014342  026523  047516  040051
2063                                        ;
2064                                        ;
2065  014350  046045  047105  052107       CON6:   .ASCII  /%LENGTH (1 TO SWRDCT)?@/
2066  014356  020110  030450  052040
2067  014364  020117  053523  042122
2068  014372  052103  037451     100
2069                                        ;
2070                                        ;
2071  014377     045  052123  051101       CON7:   .ASCII  /%STARTING SURFACE AND SECTOR?@/
2072  014404  044524  043516  051440
2073  014412  051125  040506  042503
2074  014420  040440  042116  051440
2075  014426  041505  047524  037522
```

```
2076  014434     100
2077                                    ;
2078                                    ;
2079  014435     045  042523  042514  CON7A:  .ASCII  /%SELECT CYLINDER FROM SR0-77 (YES-NO)?/
2080  014442  052103  041440  040131
2081  014450  047111  042504  020122
2082  014456  051106  046517  051440
2083  014464  030122  033455  020077
2084  014472  054450  051505  047055
2085  014500  024517     100
2086                                    ;
2087  014503     045  050117  044524  CON8:   .ASCII  /%OPTIONAL DATA PATTERN NO.?/
2088  014510  047117  046101  042040
2089  014516  052101  020101  040520
2090  014524  052124  051105  020116
2091  014532  047516  037456     100
2092                                    ;
2093                                    ;
2094  014537     045  051127  052111  CON9:   .ASCII  /%WRITE?(YES-NO)?/
2095  014544  037505  054450  051505
2096  014552  047055  024517     100
2097                                    ;
2098                                    ;
2099  014557     045  051127  052111  CON10:  .ASCII  /%WRITE CHECK?(YES-NO)?/
2100  014564  020105  044103  041505
2101  014572  037513  054450  051505
2102  014600  047055  024517     100
2103                                    ;
2104                                    ;
2105  014605     045  042522  042101  CON11:  .ASCII  /%READ?(YES-NO)?/
2106  014612  024077  042531  026523
2107  014620  047516  040051
2108                                    ;
2109  014624  051045  040505  020104  CON12:  .ASCII  /%READ CHECK?(YES-NO)?/
2110  014632  044103  041505  037513
2111  014640  054450  051505  047055
2112  014646  024517     100
2113                                    ;
2114                                    ;
2115  014651     040  020040     100  PAMES:  .ASCII  /  ?/
2116                                    ;
2117                                    ;
2118  014655     045  040520  052124  PAHED:  .ASCII  /%PATTERN ?/
2119  014662  051105  020116     100
2120                                    ;
2121                                    ;
2122                                    ;
2123  014667     045  047105  040104  END:    .ASCII  /%END?/
2124                                    ;
2125                                    ;
2126  014674  042045  044922  042526  NODRV:  .ASCII  /%DRIVE ZERO OFF LINE ?/
2127  014702  055040  051105  020117
2128  014710  043117  020106  044514
2129  014716  042516  040040
```

```
2130                                        ;
2131                                        ;
2132  014722  020040  042040  044522  RKNUM:  .ASCII  /  DRIVE(S) ON LINE #/
2133  014730  042526  051450  020051
2134  014736  047117  046040  047111
2135  014744  020105     100
2136                                        ;
2137                                        ;
2138          014750                        .EVEN
2139
```

```
2140
2141                                    ;
2142                                    ;
2143                                    ;
2144                                    ;
2145  014750  000000                    OUTBUF:0
2146
2147          000001                            .END
```

DZRKBA,BIC SYMBOL TABLE

```
ACNVX   013032        ACVN    012764        ADAM    011256        ADERR   002054
ADHGT   002626        ADRD    005634        ADRREP  007256        ADTST   002172
ADT1    002300        ADT2    002730        ADT2UX  003364        ADT2X   003132
ADT3    005144        ADT3UX  004070        AISRY   004374        AKH     010062
ALPHA   013044        ASKWC   001574        AUX2    003602        AU2LPA  003744
A2XSEK  003510        BALD1X  011230        BIT0  = 000001        BIT1  = 000002
BIT10 = 002000        BIT11 = 004000        BIT12 = 010000        BIT13 = 020000
BIT14 = 040000        BIT15 = 100000        BIT2  = 000004        BIT3  = 000010
BIT4  = 000020        BIT5  = 000040        BIT6  = 000100        BIT7  = 000200
BIT8  = 000400        BIT9  = 001000        BLAD1   011240        BLNK    011104
BLOCK   001102        BLSZ    010404        BUF     001056        BUFEXI  010052
BUFINX  010100        CHKADT  002700        CHKDOS  006530        CHKRA3  004330
CHOK    013176        CKCH    013146        CKHRDE  007340        CLRSUR  010220
CLRTAB  001134        CMDAE   010044        COMDAR  007754        COMPAR  011040
CONM    001412        CONTIN  010162        CONV    012752        CON1    014143
CON10   014557        CON11   014605        CON12   014624        CON2    014170
CON3    014230        CON4    014271        CON5    014312        CON6    014350
CON7    014377        CON7A   014435        CON8    014503        CON9    014537
CTBUSY  012334        DATAT   005410        DATATX  005404        DATTES  001534
DECBLK  007700        DELMES  007112        DISBUF  007624        DISK    006046
DKBUSY  012132        DKINT   006770        DMA     001052        DOWN    012142
DRVERQ  006574        DSKDR   001466        DSKNOR  001076        DSKNOS  010334
DSKRD   005560        ELH     005620        EMTRP   012502        END     014067
ERADR   007776        ERCOUN  001066        ERRVEC = 000004       ERR1    002472
ERR10   003714        ERR11   004020        ERR12   004240        ERR13   004360
ERR14   004636        ERR15   004736        ERR16   005036        ERR17   005254
ERR2    002620        ERR20   005374        ERR3    002674        ERR4    003124
ERR5    003206        ERR6    003306        ERR7    003632        ESW     005530
ESRDCH  005736        EXGEN   010770        EXMFLG  002264        EXRAX   006472
EXREF   011514        EXTAD2  003360        EXTDR   010332        EXTMEN  011452
EXTPP   006634        FILDAT  010630        FLAG    001042        HED1    013470
HED2    013505        HED4    013524        HED5    013554        HED5A   013552
HED6    013575        HINUM   010774        HRDER   001100        HRDEXI  007350
INBUF   001106        INCADT  002644        INCSEC  007650        INCSUR  007710
INCTAB  013426        INTEXT  007510        LCONM   001370        LDAT    005452
LGMEM   011542        LKADER  005360        LODEN   010424        LOGICA  006624
LONUM   010772        LPAD2   003214        LPAD3   005402        LPAU2X  004034
LPAU3X  004460        MAMF    013246        MAXREF  011592        MES1    013013
MES1A   013630        MES11   014003        MES12   014015        MES13   014020
MES2    013645        MES2A   013662        MES3    013677        MES4    013721
MES5    013737        MES6    013757        MES7    013777        MES8    014001
MODRTX  011730        MSTR    006030        MYWAY   011630        NBOSL   004254
NOCHA   013034        NODRV   014674        NOISET  004534        NOP   = 000240
NULL    012716        OPDAR   001716        OPDSEL  007540        OPPAT   001752
OPRD    002060        OPWCK   002134        OPWRT   002032        OUTBUF  014750
PARCSR = 172100       PARERR  014056        PARSRV  013340        PARVEC = 000114
PASEL   010552        PASS    001074        PASSC   001104        PATHED  014055
PATMES  014651        PATNU   001054        PAT0    010770        PAT1    011000
PAT10   011016        PAT11   011020        PAT12   011022        PAT13   011024
PAT14   011026        PAT15   011030        PAT16   011032        PAT17   011034
PAT2    011002        PAT20   011036        PAT3    011004        PAT4    011006
PAT5    011010        PAT6    011012        PAT7    011014        POLOK   003324
PRIORI  001040        PS      001002        PWAIT   005116        PWRF    014103
RANDOM  010644        RANER   006416        RANEX   006104        RANNU   001044
```

DZRKBA,BIC SYMBOL TABLE

```
RCHWAI  005726        RDCHK   002106        RDCK    = 104413        RDCKLP  005066
RDSECT  002544        RDWAIT  005610        READ    = 104405        REDA    007144
REPDRV  001304        REPOEN  006604        RKBA    001020          RKCS    001014
RKDA    001022        RKDB    001030        RKDS    001026          RKER    001024
RKMR    001032        RKNODV  001270        RKNUM   014722          RKWC    001016
RORBLK  010470        RTIX    012040        SANHT   002666          SAVE    001070
SAV1    001072        SAWTH   004472        SAW1    004654          SAW2    004754
SAW3    005054        SEABUF  002376        SEEKCY  004150          SELCMA  012052
SELDAR  012062        SELDCS  012072        SELWC   012042          SETADT  002512
SETBLK  010432        SETPAS  011740        SHIFT   010692          SLW     005744
SOFTER  007444        SP      =%000006      SPECME  014125          SPECOR  001234
SRCHG   012344        STAER   012356        STAER1  012422          START   001120
STATUS  001036        SWR     001000        SWRDCT  001064          SWSUP   007744
TABLE   013450        TDMA    001062        TEXBUF  013244          TIMQ    014030
TKB     001006        TKS     001012        TKSEL   012212          TKSR    001070
TPB     001004        TPS     001010        TRACK   001050          TSRDY   003060
TSTAUX  003610        TSTCH   013424        TSTFLG  013064          TSTNXS  005262
TSTRA3  004210        TSTZER  005206        TWRDCT  001060          TYPXIT  012064
TYP     012514        TYPA    012524        TYPC    012552          TYPD    012600
TYPDAT  012714        TYPF    012666        TYPG    012700          TYPS    012720
TYPSA   012744        TYPSB   012746        TYST    013052          UP      012166
VECTOR  001034        WCCON   001614        WCWAIT  006012          WDERR   011260
WORK    001110        WORK1   001112        WORK2   001114          WORK3   001116
WRAUSK  003446        WRCHEC  = 104407      WRDCMP  011126          WRDCT   001040
WRDINC  011136        WRITE   = 104403      WRLG    006154          WRNEXB  002500
WRWAIT  005512        XESH    006022        XINCSE  010134          XINCSU  010172
XINCW   011776        XSEABU  002422        XSLH    005522          XWAIT   011714
XWPAT   005136        .       = 014752
```

ERRORS DETECTED: 0

DZRKBA,BIC

```
*DZRKBA,DZRKBA/SOL=DZRKBA,BIC
RUN-TIME:  9 19 .1 SECONDS
CORE USED:  4K
```

digital

MAINDEC CHANGE NOTICE

CHANGE NO.
11-D5HA- 4
Sheet 1 of 1

AUTHOR	PROGRAM DATE	PRODUCT LINE	MAINDEC NUMBER
E. Haight DATE EXT. 4/12/71 3246	APRIL 71	PDP-11	11-D5HA

PROGRAM NAME RK11 Disk Data DEVICE RK11

ITEM

1.

PROBLEM: A. Starting address of power fail tests incorrect.
B. Power fail tests failed to restart when in the power up sequence.
CORRECTION: Change document section 4.6 to read 66Ø instead of 644
664 instead of 646

CHANGES TO	DATA TEST		
11754 = 24Ø	12116 = Ø12737	12124 = 24Ø	117Ø4/ ØØ2ØØØ
11756 = 24Ø	1212Ø = Ø12134	12126 = 24Ø	
1176Ø = 24Ø	12122 = 24	1213Ø = 24Ø	

2.
/3/71

PROBLEM: Overrun error occurs while operating on a 16K machine.
CORRECTION:

CHANGE	TO	CHANGE	TO	CHANGE	TO
7754	137	13464	13474	1ØØ46	42737
7756	13452	13466	137	1ØØ52	13474
13452	13737	1347Ø	7762		
13454	1Ø52	1ØØ26	32737		
13456	111Ø	1ØØ32	13474		
1346Ø	13737	1ØØ36	52737		
13462	1Ø52	1ØØ42	13474		

3.
12/71

PROBLEM: Unable to sequence through three or more drives.
CORRECTION:

CHANGE	TO	CHANGE	TO
6510	137	11754	5
6512	11754	11756	137
		1176Ø	2112

4.
12/71

PROBLEM: Program halts at location 1132Ø on 28K machine.
CORRECTION: Depress continue when halt occurs.

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-D5HA
PRODUCT NAME:	RK11 DISK DATA
DATE CREATED:	12 APRIL 1971
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	EARL HAIGHT

COPYRIGHT © 1971
DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

SECTION	CONTENTS
1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.	LOADING PROCEDURE
3.1	METHOD
4.	STARTING PROCEDURE
4.1	WORST CASE OPERATION
4.2	OPERATOR INTERVENTION
4.2.1	CHANGE DISK OPERATING PARAMETERS
4.3	SCOPE LOOP ENTRY POINTS
5.	OPERATING PROCEDURE
5.1	CONTROL SWITCH SETTINGS
5.2	CONVERSATION MODE
5.3	SUBROUTINE ABSTRACT
6.	ERROR REPORTS
7.	
8.	MISCELLANEOUS
8.1	SUGGESTED POWER FAIL TEST

1. ABSTRACT

THE RK11 DISK DATA TEST IS A SERIES OF ADDRESS AND DATA
RELIABILITY ROUTINES WHICH VERIFY TO THE USER THE DISK
CONTROL (RK11) AND DISK (RK02 THRU RK05) ARE OPERATING CORRECTLY.
THIS TEST USED IN CONJUNCTION WITH THE RK11 STATIC TEST
AND RK11 RANDOM DRIVE EXERCISER ASSURES THE USER OF
AN ERROR FREE SYSTEM, WHEN USED IN ITS ENTIRETY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11
RK11 AND (RK02 THRU RK05)

2.2 STORAGE

DATA TEST
MAIN BODY OF PROGRAM OCCUPIES ALL OF MEMORY
THE WRITE AND READ BUFFERS FLOAT THROUGH MEMORY.

3. LOADING PROCEDURE

3.1 METHOD OF LOADING DISK DATA TEST TAPE

PROGRAM FORMAT ABSOLUTE

A. VERIFY THE BOOT LOADER IS IN MEMORY.

B. SET SWITCH REGISTER EQUAL TO #750

MEMORY SIZE	#
4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

C. DEPRESS LOAD ADDRESS.

D. DEPRESS START.

4. STARTING PROCEDURE

4.1 WORST CASE TEST DRIVE ZERO FOR STATIC TEST

A) TURN ALL DRIVES EXCEPT DRIVE 0 OFF LINE.
B) SET SWITCH REGISTER EQUAL TO 200
C) DEPRESS LOAD ADDRESS
D) SET SWITCH REGISTER EQUAL TO ZERO
E) DEPRESS START

4.2 STARTING PROCEDURE COMPLETE DATA TEST

A) SET SWITCH REGISTER EQUAL TO 200
B) DEPRESS LOAD ADDRESS
C) SET SWITCH REGISTER EQUAL TO ZERO
D) DEPRESS START
E) THE PROGRAM WILL THEN INQUIRE TO THE NUMBER
OF DRIVES ON THE SYSTEM, AFTER THE OPERATOR
ANWSERS THE QUESTION THE TEST WILL START RUNNING

4.3 OPERATOR INTERVENTION FOR DATA TEST ONLY.

A) SET SWITCH REGISTER EQUAL TO 200
B) DEPRESS LOAD ADDRESS
C) SET SWITCH REGISTER EQUAL TO MODE OF OPERATION (REF. SEC5.)
D) DEPRESS START.

4.4 ADDRESS ENTRY POINTS FOR STATIC TEST ONLY

300	STAI1	TEST RESET FUNCTION
304	STAI2	TEST RESET CURRENT ADDRESS REGISTER
310	STAI3	TEST RESET WORD COUNT REGISTER
314	STAI4	TEST RESET DRIVE ADDRESS REGISTER
320	STAI5	TEST RESET CONTROL ERROR REGISTER
324	STAI6	TEST RESET DATA BUFFER REGISTER
330	STAI7	TEST RESET MAINTENANCE REGISTER
334	STAI10	TEST CAN WE SET FUNCTION REGISTER BITS
340	STAI11	TEST CAN WE CLEAR FUNCTION REGISTER BITS
344	STAI12	TEST CAN WE SET ALL CMA BITS
350	STAI13	TEST CAN WE CLEAR ALL CMA BITS
354	STAI14	TEST CAN WE SET ALL WORD COUNT BITS
360	STAI15	TEST CAN WE CLEAR ALL WORD COUNT BITS
364	STAI16	TEST CAN WE SET ALL DRIVE ADDRESS BITS
370	STAI17	TEST CAN WE CLEAR ALL DRIVE ADRESS BITS

374	STAI22	EXECUTE A ONE WORD WRITE
400	STAI40	EXECUTE A ONE WORD WRITE CHECK
404	STAI56	EXECUTE A ONE WORD READ
410	INHCMA	TEST CURRENT ADDRESS INHIBIT
414	STAI74	TEST THE SURFACE INCREMENT
420	STAI77	TEST THE CYLINDER INCREMENT
424	STA103	TEST THE OVERRUN ERROR
430	NXDST	TEST THAT NED RAISES ERROR FLAG
434	ST103X	TEST THE NON-EXISTENT SECTOR ERROR
440	ST104X	TEST THE NON-EXISTENT CYLINDER ERROR
444	ST105X	TEST THE PROGRAM ERROR
450	NXMTSM	TEST THE NON-EXISTENT MEMORY ERROR
454	STA106	TEST THE CONTROL WILL NOT TRAP AT BR7
4o0	STA110	TEST THE CONTROL WILL NOT TRAP AT BR6
464	STA112	TEST THE CONTROL WILL NOT TRAP AT BR5
470	STA114	TEST THE CONTROL WILL TRAP AT BR4
474	RESTORE	TEST DRIVE RESTORE
500	ADT1	TEST THE DRIVE SECTOR FORMAT

4.4.1 ADDRESS ENTRY POINTS FOR TEST ROUTINES (DATA TEST ONLY)

600	ADDRESS TEST 1
604	ADDRESS TEST 2
610	ADDRESS TEST 3
614	ADDRESS TEST 4
620	ADDRESS TEST 5
624	ADDRESS TEST 6
630	RANDOM DRIVE EXERCISER

4.5 ***MAINTENANCE ROUTINES*** (BOTH STATIC AND DATA TESTS)

634	SELWC	LOAD WORD COUNT REG. WITH SWITCH REGISTER
640	SELCMA	LOAD CURRENT MEMORY ADDR REG. WITH SWITCH REGISTER
644	SELDAR	LOAD DISK ADDR. REGISTER WITH SWITCH REGISTER
650	SELDCS	LOAD DISK CONTROL REGISTER WITH SWITCH REGISTER
654	TKSEL	LOAD CYLINDER ADDRESS REGISTER FROM SWITCH REGISTER AND SEEK.

4.6 POWER FAIL TESTS (DATA TEST ONLY)

644	PFT1	TEST POWER FAIL DURING WRITE
646	PFT2	TEST POWER FAIL DURING A READ

5. OPERATING PROCEDURE

5.1 CONTROL SWITCH SETTINGS

PROGRAM CONVERSATION

SR15	SET	ENTER PROGRAM CONVERSATION MODE REF. SEC. 5.2
	RESET	OPERATE WORSE CASE DISK ZERO

DELETE TYPEOUT

SR14	SET	DELETE TYPEOUTS
	RESET	REPORT MESSAGE

HALT ON READY FLAG

SR13	SET	HALT ON FLAG (READY SET)
	RESET	EXECUTE NEXT OPERATION

(5,1 CONT'D)

DELETE COMPARISIONS

SR12 SET DELETE DATA COMPARISONS

RESET COMPARE DATA BUFFERS

LOOP ON TEST

SR11 SET LOOP ON TEST

RESET CONTINUE TO NEXT TEST

HALT ON ERROR

SR10 SET HALT AFTER ERROR REPORT

RESET CONTINUE AFTER ERROR REPORT

ENTER INTERRUPT BACKGROUND TEST

SR9 SET WAIT FOR INTERRUPTS USING WAIT INSTRUCTION

RESET WAIT FOR INTERRUPTS WITH BACKGROUND TEST

LOOP ON DISK ADDRESS

SR8 SET LOOP ON DISK ADDRESS (SPECIFIED BY PARAMETERS)

RESET CONTINUE TO NEXT DISK BUFFER AREA.

SR7-Ø SELECT CYLINDER FROM SR (DURING DYNAMIC TESTING)

CYLINDER SELECTION

7	6	5	4	3	2	1	Ø
128	64	32	16	8	4	2	1

SELECT ONE OF 312(8) CYLINDERS

5.2 CONVERSATION MODE FOR PROGRAM PARAMETERS FOR DATA TEST ONLY

IN THE PROGRAM CONVERSATION MODE THE OPERATOR CAN SPECIFY ANY ONE OR ALL OF THE PROGRAM PARAMETERS.

```
                PROGRAM CONVERSATION

        DATA TEST ONLY?
```

IF THE OPERATOR ANSWER YES THE PROGRAM WILL ENTER ONLY THE DATA PORTION OF TEST.

```
        MULTI DRIVE MODE?
```

MULTI DRIVE MODE IS A MODE IN THE PROGRAM WHICH ALLOWS THE OPERATOR TO EXECISE ALL THE DRIVES ON THE SYSTEM WITHOUT RE-STARTING THE PROGRAM. THE PROGRAM AFTER EXERCISING ONE DRIVE WILL REPORT A MESSAGE TELLING THE OPERATOR WHICH DRIVE WILL BE SELECTED NEXT, AND THEN THE PROGRAM WILL EXERCISE THAT DRIVE. WHEN A COMPLETE PASS IS ACCOMPLISHED, A PASS COMPLETE WILL BE REPORTED AND THE TEST WILL RECYCLE.

IF THE OPERATOR ANSWERS "YES" TO THIS QUESTION, HE WILL THEN BE ASKED HOW MANY DRIVES ARE ON THE SYSTEM, AND THEN THE PRECEDING QUESTION WILL BE SKIPPED. IF THE OPERATOR ANSWERS "NO" TO THIS QUESTION, THE NEXT QUESTION WILL BE SKIPPED, AND HE WILL THEN BE ASKED WHICH DRIVE IS TO BE EXERCISED.

```
        NUMBER OF DRIVES 1 TO 10 OCTAL?X
```

TYPE THE NUMBER OF DRIVES ON THE SYSTEM, FOR MULTI DISK MODE.

```
        EXERCISE DRIVE?X
```

WHEN NOT IN THE MULTI DRIVE MODE THE OPERATOR WILL HAVE TO SPECIFY WHAT DRIVE IS TO BE USED.

```
        OPTIONAL WORD COUNT(YES-NO)?X
```

IF THE OPERATOR ANSWERS "NO" TO THIS QUESTION THE NEXT QUESTION WILL BE DELETED FROM THE CONVERSATION.

```
        LENGTH (1 TO SWRDCT)?  XXX
```

THE OPERATOR CAN SPECIFY ANY LENGTH TRANSFER FROM 1(8) TO SWRDCT, THE NORMAL TRANSFER LENGTH VARIES FOR EACH SYSTEM THE NORMAL WORD COUNT (SWRDCT) IS REPORTED EACH TIME THE PROGRAM IS STARTED.

```
        SELECT CYLINDER FROM SR7=0?(YES=NO)
```

IF THE OPERATOR WISHES HE MAY SELECT THE CYLINDER ADDRESS VIA THE SWITCH REGISTER WHEN RUNNING THE DATA TEST BY ANWSERING YES TO THIS QUESTION.

```
        STARTING SURFACE AND SECTOR? X
```

IF THE OPERATOR WISHES TO START A TRANSFER AT A PARTICULAR SURFACE AND SECTOR HE MAY BY SPECIFYING THOES PARAMETERS HERE.

*THE MAX. VALUE FOR THE SURFACE AND SECTOR MAY NOT
EXCEED 33(8). THE SECTORS ON SURFACE ZERO RUN
FROM 0 TO 13(8). THE SECTORS ON SURFACE ONE RUN FROM
20(8) TO 33(8).

(5,2 CONT'D)

OPTIONAL DATA PATTERN NO,?X

IF NO OPTIONAL DATA PATTERN IS REQUESTED THE PROGRAM WILL EXECUTE THE FOLLOWING LIST OF DATA PATTERNS,

PATTERN 0 = 000000
" 1 = 177777
" 2 = 134510
" 3 = 043267
" 4 = 100000
" 5 = 107070
" 6 = 070707
" 7 = 052525
" 10 = 125252
" 11 = 177737
" 12 = 004102
" 13 = 136363
" 14 = 063636
" 15 = 000001
" 16 = 100005
" 17 = 000520
" 20 = 030303
" 21 = RANDOM DATA
" 22 = RUN ALL DATA PATTERNS UNDER PROGRAM CONTROL

IF THE OPERATOR DECIDES TO OPERATE UNDER PROGRAM CONTROL, THE NEXT QUESTION WILL BE SKIPPED. THE OPERATOR BY REQUESTING AN OPTIONAL DATA PATTERN HAS THE ABILITY OF SELECTING ANY ONE OF THE 21(8) DATA PATTERNS. DATA PATTERN NO. XX

CONTROL PROGRAM FUNCTION PARAMETERS

IN THIS SECTION OF THE PROGRAM PARAMETER CONVERSATION MODE, THE OPERATOR CAN SELECT ANYONE OR ALL THREE OF THE CONTROL FUNCTIONS TO BE EXECUTED. THE NORMAL SEQUENCE OF CONTROL FUNCTIONS UNDER PROGRAM CONTROL ARE WRITE, READ, READ CHECK AND WRITE CHECK. BY ENTERING THE CONVERSATION MODE THE OPERATOR HAS GAINED COMPLETE CONTROL OVER THE CONTROL FUNCTIONS. HE MUST SPECIFY YES OR NO TO ALL OF THE FOLLOWING QUESTIONS,

WRITE? (YES - NO)
READ? (YES - NO)
READ CHECK? (YES - NØ)
WRITE CHECK? (YES - NO)

TO PERFORM A WRITE CHECK ONLY, THE OPERATOR MUST FIRST WRITE SOME KNOWN DATA ON THE DISK. THIS COURSE OF ACTION ALSO PREVAILS FOR A READ ONLY OPERATION.

* IF AN ERROR OCCURS IN THE LINE THE OPERATOR IS TYPING, DEPRESS THE RUB-OUT. THIS CAUSES THE QUESTION TO BE RETYPED AND ALLOWS THE OPERATOR TO PROPERLY ANSWER THE QUESTION,

* INDICATES TO THE OPERATOR THAT THE CARRIAGE-RETURN SHOULD BE TYPED AT THE INDICATED PLACE TO TERMINATE THE LINE OF TYPED CHARACTERS,

TKSEL - STATIC CYLINDER SELECTION

THIS ROUTINE WAS DESIGNED TO ENABLE THE OPERATOR TO HAVE A QUICK METHOD OF SELECTING CYLINDER AND ISSUING A SEEK.

5.3 SUBROUTINE ABSTRACTS (STATIC TEST ONLY)

THERE ARE NO ABSTRACTS FOR THE STATIC TEST BECAUSE, THE PROGRAM CONSIST OF MANY SMALL TESTS WHICH HAVE BREIF DESCRIPTION AT THE BEGINNING OF EACH ONE.

5.3.1 SUBROUTINE ABSTRACTS (DATA TEST ONLY)

ADDRESS TEST 1 (ADT1)

IN THIS TEST THE PROGRAM WRITES EACH SECTOR ON ITSELF AND THEM READS IT BACK AND VERIFIES THAT THE CORRECT SECTOR WAS READ. THIS TEST USES A BLOCK LENGTH OF 400 WORDS.

ADDRESS TEST 2 (ADT2)

IN THIS TEST THE ABILITY OF THE CONTROL AND DRIVE TO SEEK FROM CYLINDER 0 TO EACH OTHER CYLINDER IS CHECKED. ONLY SEEKS ARE USED NOT RESTORES.

ADDRESS TEST 3 (ADT2UX)

THIS THIS TEST THE CYLINDER SELECTION MATRIX IS CHECKED BY WRITING THE FIRST WORD OF EACH CYLINDER WITH ITS OWN CODE,THEN THE PROGRAMS STARTS SEEKING FROM CYLINDER 312 TO ALL OTHER CYLINDERS TO VERIFY CORRECT SELECTION.

ADDRESS TEST 4 (ADT3UX)

IN THIS TEST THE ABILITY OF THE DRIVE TO DO MAX. HEAD MOVEMENT IS CHECKED. IN THIS THE DRIVE FIRST SEEKS TO 0 THEN TO CYLINDER 312, THEN BOTH THE COUNTS ARE ALTERED INSTEAD OF SEEKING TO 0 THIS TIME WE SEEK TO 1,AND INSTEAD SEEKING TO 312 WE SEEK TO 311, THIS CONTINUES UNTIL 0 INCREMENTS TO 312,AND THE COUNT THAT START AT 312 GOES TO 0.

ADDRESS TEST 5 (SAWTH)

IN THIS TEST THE PROGRAM CREATES HEAD MOVEMENT WHICH APPEARS AS A SAW TOOTH PATTERN, THIS IS TO CREAT A WORST CASE NOISE.

ADDRESS TEST 6 (LOOK AHEAD)

IN THIS TEST THE PROGRAM CHECKS THE ABILITY OF THE DRIVE TO LOCATE THE SECTOR ADDRESS .

RANDOM DRIVE EXERCISER (RANEX)

IN THIS TEST THE ABILITY OF THE DRIVE TO DO RANDOM SEEKS AND TO WRITE AND READ RANDOM DATA AT THE SAME TIME IS CHECKED. THE WORD COUNT FOR THIS TEST IS 400 WORDS OR ONE BLOCK.

(5,3 CONT'D)

TKSEL - OPERATING PROCEDURE

STEP A. SET SWITCH REGISTER EQUAL TO 654

STEP B. DEPRESS LOAD ADDRESS

STEP C. SET SWITCH REGISTER 10 THRU 8 EQUAL TO DRIVE #0 THRU 7

STEP D. DEPRESS START

STEP E. SET SWITCH REGISTER 7 THRU 0 EQUAL TO CYLINDER #

STEP F. DEPRESS CONTINUE

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

CYLINDER NUMBER
(0 THRU 177(8))

STEP G. TRACK NUMBER CAN BE CHANGED ARBITRARILY.

6. ERROR REPORTS

6.1 STATIC AND ADDRESS REPORT.

XX ERR CNT XXXXXXWRD1XXXXWRD2

ERRCNT = IS THE TAG FOR THE LISTING
WRD1 = WHAT WAS EXPECTED
WRD2 = WHAT WAS RECEIVED

WHEN A REPORT ONLY CONTAINS ONE WORD THE PROGRAM WAS EXPECTING ZEROS BUT RECEIVED WHAT WAS REPORTED.

6.2 ERROR REPORTS

STATUS ERROR

STATUS ERROR XXXXXXRKDA XXXXXXRKER XXXXXXRKCS

A B C

A=THE DRIVE ADDRESS REGISTER
B=THE CONTROL ERROR REGISTER
C=THE CONTROL FUNCTION REGISTER

BIT LAYOUT OF RKCS REGISTER

BIT15= ERROR
BIT14= HARD ERROR
BIT13= SEARCH COMPLETE
BIT12= MAINTENANCE ENABLE
BIT11= NOT USED
BIT10= FORMAT
BIT9 = READ/WRITE ALL
BIT8 = STOP ON SOFT ERROR
BIT7= READY
BIT6= INTERRUPT ENABLE
BIT5= EXTENDED MEMORY 1 (XM1)
BIT4= EXTENDED MEMORY 0 (XM0)
BIT3-2-1= FUNCTION REGISTER
BIT0= GO

BIT 3	BIT 2	BIT 1	OPERATION
0	0	0	RESET
0	1	0	READ
0	0	1	WRITE
0	1	1	WRITE CHECK
1	0	0	SEEK
1	0	1	READ CHECK
1	1	0	DRIVE RESET
1	1	1	WRITE LOCK

BIT LAYOUT OF RKER REGISTER

BIT15= DRIVE ERROR (DRE)
BIT14= OVERRUN (OVR)
BIT13= WRITE LOCK OUT VIOLATION (WLO)
BIT12= SEEK ERROR (SKE)
BIT11= PROGRAMMING ERROR (PCE)
BIT10= NON-EXISTENT MEMORY ERROR (NXM)
BIT9 = DATA LATE ERROR (DTL)
BIT8 = READ TIMING ERROR (RTE)
BIT7 = NON-EXISTENT DISK ERROR (NXD)
BIT6 = NON-EXISTENT CYLINDER ERROR (NXC)
BIT5 = NON-EXISTENT SECTOR (NXS)
BIT4-3-2= NOT USED
BIT1 = CHECKSUM ERROR (CSE)
BIT0 = WRITE CHECK ERROR (WCE)

IF THE ERROR OCCURRED WHEN READING THE PROGRAM WILL REPORT WHICH READ.

NOTE: WHEN A HARD ERROR OCCURS AN ADDITIONAL ERROR MESSAGE
WILL BE REPORTED, AS FOLLOWS:

XXXXXX RKDS

LAYOUT OF RKDS BITS

BIT15-14-13= IDENTIFICATION OF DRIVE
BIT12= DRIVE POWER LOW (DPL)
BIT11= HIGH DENSITY DRIVE (HDEN)
BIT10= DRIVE UNSAFE (DRU)
BIT9 = SEEK INCOMPLETE (SIN)
BIT8 = SECTOR COUNTER OK (SOK)
BIT7 = DRIVE READY (DRY)
BIT6 = ACCESS READY (ARDY)
BIT5 = WRITE PROTECT STATUS (WPS)
BIT4 = SECTOR COUNTER EQUALS SECTOR ADDRESS (SC=SA)
BIT3-0= SECTOR COUNTER (S.C.)

6.3 DATA ERRORS

```
 1 READ  XXXXXXRKDA XXXXXXGD DATA XXXXXXBD DATA

   A         B             C             D
```

A=WHICH READ THE ERROR OCCURRED
B=THE DRIVE ADDRESS REGISTER
C=THE DATA WRITTEN ON THE DRIVE SURFACE
D=THE DATA READ FROM THE DRIVE SURFACE

6.4 RANEX ERRORS

ERRORS WHICH OCCUR IN THE RANDOM EXERCISER (RANEX) WILL ALSO HAVE THE FOLLOWING MESSAGE REPORTED.

```
RANEX ERROR
```

6.7 PROCESSOR TIME OUT

```
CPU BKGRND TIMED OUT
```

THIS MESSAGE WILL BE REPORTED IF THE RK11 CONTROL FAILS TO RAISE A BR REQUEST AFTER EXTENDED PERIOD OF TIME.

6.8.1 MULTIPLE DRIVE OPERATION

```
NUMBER OF DRIVES 1 TO 10 OCTAL?XX
```

THIS MESSAGE IS REPORTED AT THE START OF THE PROGRAM WHEN BIT15 IS NOT SET IN THE DATA TEST. THE NUMBER THAT THE OPERATOR TYPES HERE INDICATES THE NUMBER OF DRIVES THE PROGRAM WILL AUTOMATICALLY SEQUENCE THROUGH ON THE SYSTEM.

NOTE: THE DRIVES MUST BE IN NUMERICAL SEQUENCE.

6.8.2 STANDARD WORD COUNT REPORT (SWRDCT)

```
XXXXXXSWRDCT
```

THIS MESSAGE IS REPORTED AT THE START OF THE PROGRAM TO TELL THE OPERATOR WHAT THE LARGEST TRANSFER IS THAT HE MAY MAKE TO THE RK11 DRIVE. THIS WORD COUNT VARIES FROM SYSTEM TO SYSTEM SINCE THE BUFFERS FLOAT THROUGH MEMORY.

6.8 END

END

THIS MESSAGE IS REPORTED AT THE END OF ONE COMPLETE PASS OF THE DISK SYSTEM.

8. MISCELLANEOUS

IN SOME ADDRESS TESTS THE PROGRAM DEPENS ON WRITTING AND READING DATA CORRECTLY FROM THE DISK,AND IF IT DOES NOT IT MAY REPORT AN ADDRESS FAILURE. WHEN IN FACT IT WAS A DATA FAILURE.

```
                000000                  N=0
                000001                  BIT0=1
                000002                  BIT1=2
                000004                  BIT2=4
                000010                  BIT3=10
                000020                  BIT4=20
                000040                  BIT5=40
                000100                  BIT6=100
                000200                  BIT7=200
                000400                  BIT8=400
                001000                  BIT9=1000
                002000                  BIT10=2000
                004000                  BIT11=4000
                010000                  BIT12=10000
                020000                  BIT13=20000
                040000                  BIT14=40000
                100000                  BIT15=100000
                000004                  SCOPE=IOT
                                        ;
                                        ;
                                        ;
                104403                  WRITE=TRAP+3
                104407                  WRCHECK=TRAP+7
                104405                  READ=TRAP+5
                000000                  .=0

                                        ;MACRO FOR SETTING UP ERROR COUNT
                                                .MACR   ERROR X
                                        ERR'X'; MOV     #X,ERCOUNT      ;SET UP ERROR COUNT X
                                                N=N+1
                                                .ENDM
                000100                  .REPT   100
                                        .+2
                                        HALT
                                        .ENDR
000000  000002                          .+2
000002  000000                          HALT
000004  000006                          .+2
000006  000000                          HALT
000010  000012                          .+2
000012  000000                          HALT
000014  000016                          .+2
000016  000000                          HALT
000020  000022                          .+2
000022  000000                          HALT
000024  000026                          .+2
000026  000000                          HALT
000030  000032                          .+2
000032  000000                          HALT
000034  000036                          .+2
000036  000000                          HALT
000040  000042                          .+2
000042  000000                          HALT
000044  000046                          .+2
```

```
000046  000000                  HALT
000050  000052                  ,+2
000052  000000                  HALT
000054  000056                  ,+2
000056  000000                  HALT
000060  000062                  ,+2
000062  000000                  HALT
000064  000066                  ,+2
000066  000000                  HALT
000070  000072                  ,+2
000072  000000                  HALT
000074  000076                  ,+2
000076  000000                  HALT
000100  000102                  ,+2
000102  000000                  HALT
000104  000106                  ,+2
000106  000000                  HALT
000110  000112                  ,+2
000112  000000                  HALT
000114  000116                  ,+2
000116  000000                  HALT
000120  000122                  ,+2
000122  000000                  HALT
000124  000126                  ,+2
000126  000000                  HALT
000130  000132                  ,+2
000132  000000                  HALT
000134  000136                  ,+2
000136  000000                  HALT
000140  000142                  ,+2
000142  000000                  HALT
000144  000146                  ,+2
000146  000000                  HALT
000150  000152                  ,+2
000152  000000                  HALT
000154  000156                  ,+2
000156  000000                  HALT
000160  000162                  ,+2
000162  000000                  HALT
000164  000166                  ,+2
000166  000000                  HALT
000170  000172                  ,+2
000172  000000                  HALT
000174  000176                  ,+2
000176  000000                  HALT
000200  000202                  ,+2
000202  000000                  HALT
000204  000206                  ,+2
000206  000000                  HALT
000210  000212                  ,+2
000212  000000                  HALT
000214  000216                  ,+2
000216  000000                  HALT
000220  000222                  ,+2
000222  000000                  HALT
```

```
000224  000226                    .+2
000226  000000                    HALT
000230  000232                    .+2
000232  000000                    HALT
000234  000236                    .+2
000236  000000                    HALT
000240  000242                    .+2
000242  000000                    HALT
000244  000246                    .+2
000246  000000                    HALT
000250  000252                    .+2
000252  000000                    HALT
000254  000256                    .+2
000256  000000                    HALT
000260  000262                    .+2
000262  000000                    HALT
000264  000266                    .+2
000266  000000                    HALT
000270  000272                    .+2
000272  000000                    HALT
000274  000276                    .+2
000276  000000                    HALT
000300  000302                    .+2
000302  000000                    HALT
000304  000306                    .+2
000306  000000                    HALT
000310  000312                    .+2
000312  000000                    HALT
000314  000316                    .+2
000316  000000                    HALT
000320  000322                    .+2
000322  000000                    HALT
000324  000326                    .+2
000326  000000                    HALT
000330  000332                    .+2
000332  000000                    HALT
000334  000336                    .+2
000336  000000                    HALT
000340  000342                    .+2
000342  000000                    HALT
000344  000346                    .+2
000346  000000                    HALT
000350  000352                    .+2
000352  000000                    HALT
000354  000356                    .+2
000356  000000                    HALT
000360  000362                    .+2
000362  000000                    HALT
000364  000366                    .+2
000366  000000                    HALT
000370  000372                    .+2
000372  000000                    HALT
000374  000376                    .+2
000376  000000                    HALT
```

```
        000200                  .=200
000200  000167  000676         JMP     START

                                ;STATIC ROUTINES
        000300                  .=300
000300  000167  000706                 JMP     STAI1           ;TEST RESET TO CONTROL REGISTER
000304  000167  000760                 JMP     STAI2           ;TEST RESET TO CURRENT ADDRESS REGISTER
000310  000167  001004                 JMP     STAI3           ;TEST RESET TO WORD COUNT REGISTER
000314  000167  001030                 JMP     STAI4           ;TEST RESET TO DISK ADDRESS REGISTER
000320  000167  001054                 JMP     STAI5           ;TEST RESET TO CONTROL ERROR REGISTER
000324  000167  001100                 JMP     STAI6           ;TEST RESET TO DATE BUFFER REGISTER
000330  000167  001124                 JMP     STAI7           ;TEST RESET TO MAINTENANCE REGISTER
000334  000167  001150                 JMP     STAI10          ;CAN WE SET W/R BITS IN DCS REGISTER
000340  000167  001222                 JMP     STAI11          ;CAN WE CLEAR THE DCS REGISTER
                                                               ;USING DISK CLEAR.
000344  000167  001302                 JMP     STAI12          ;CAN WE SET ALL CMA BITS
000350  000167  001346                 JMP     STAI13          ;CAN WE CLEAR ALL CMA BITS
                                                               ;USING DISK CLEAR
000354  000167  001416                 JMP     STAI14          ;CAN WE SET ALL WC BITS
000360  000167  001462                 JMP     STAI15          ;CAN WE CLEAR ALL WC BITS
                                                               ;USING DISK CLEAR
000364  000167  001526                 JMP     STAI16          ;CAN WE SET ALL THE RKDA BITS
000370  000167  001572                 JMP     STAI17          ;CAN WE CLEAR ALL THE DAR BITS
                                                               ;USING DISK CLEAR
```

```
000374  000167  001642                JMP     STAI22          ;EXECUTE A ONE WORD WRITE
                                                              ;FOLLOWED BY A ONE WORD WRITE CONTINUE
000400  000167  002510                JMP     STAI40          ;EXECUTE A ONE WORD WRITE CHECK
                                                              ;FOLLOWED BY A ONE WORD WRITE CHECK CONTINUE
000404  000167  003352                JMP     STAI56          ;EXECUTE A ONE WORD READ
                                                              ;FOLLOWED BY A ONE WORD READ CONTINUE
000410  000167  004220                JMP     INHCMA          ;TEST CURRENT ADDRESS INHIBT
000414  000167  004362                JMP     STAI74          ;TEST SURFACE INCREMENT
000420  000167  004524                JMP     STAI77          ;TEST CYLINDER INCREMENT
000424  000167  004704                JMP     STA103          ;TEST OVERRUN ERROR
000430  000167  005052                JMP     NXDST           ;TEST THAT NED RAISES ERROR FLAG
000434  000167  005230                JMP     ST103X          ;TEST NON-EXISTENT SECTOR
000440  000167  005372                JMP     ST104X          ;TEST NON-EXISTENT CYLINDER
000444  000167  005534                JMP     ST105X          ;CHECK PROGRAM ERROR
000450  000167  005656                JMP     NXMTSM          ;TEST NON-EXISTENT MEMORY ERROR
000454  000167  006152                JMP     STA106          ;TEST THAT THE DISK WILL NOT TRAP
                                                              ;AT PRIORITY 7
000460  000167  006276                JMP     STA110          ;TEST THAT THE DISK WILL NOT TRAP
                                                              ;AT PRIORITY 6
000464  000167  006422                JMP     STA112          ;TEST THAT THE DISK WILL NOT TRAP
                                                              ;AT PRIORITY 5
000470  000167  006546                JMP     STA114          ;TEAST THAT THE DISK WILL TRAP
                                                              ;AT PRIORITY 4
000474  000167  006656                JMP     RESTORE         ;TEST RESTORE FUNCTION
000500  000167  007100                JMP     ADT1            ;CHECK SECTOR FORMAT
        000634                  .=634
                                ;
                                ;MAINTENANCE ROUTINES
                                ;
000634  000167  007320                JMP     SELWC           ;LOAD WORD COUNT REG WITH SR
000640  000167  007324                JMP     SELCMA          ;LOAD CURRENT ADDRESS REG WITH SR
000644  000167  007330                JMP     SELDAR          ;LOAD DISK ADDRESS REG WITH SR
000650  000167  007334                JMP     SELDCS          ;LOAD FUNCTION REG WITH SR
                                ;
                                ;
000654  000167  007400                JMP     TKSEL           ;SELECT CYLINDERS STATICLY
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;RF11 DATA TEST
                                ;VECTORS USED IN PROGRAM
                                ;#1 LOC 204 DISK INTERRUPT
                                ;#2 LOC 30 EMT (TELETYPE OUTPUT)
                                ;#3 LOC 34 TRAP (DISK HANDLERS)
                                ;
        001000                  .=1000
```

```
001000                          .EVEN
                                ;
                                ;I/O ADDRESS POINTERS
001000  177570                  SR:     177570          ;SWITCH REGIISTER
001002  177776                  CSR:    177776          ;PROCESSOR STATUS REGISTER
001004  177566                  TPB:    177566          ;TELETYPE REGISTERS
001006  177562                  TKB:    177562
001010  177564                  TPS:    177564
001012  177560                  TKS:    177560
                                ;
                                ;DISK I/O REGISTERS
                                ;
001014  177404                  DCS:    177404          ;DISK CONTROL REGISTER
001016  177406                  WC:     177406          ;WORD COUNT REGISTER
001020  177410                  CMA:    177410          ;CURRENT MEMORY ADDRESS REGISTER
001022  177412                  DAR:    177412          ;LOWER 16 BITS OF DISK ADDRESS
001024  177402                  RKER:   177402          ;CONTROL ERROR REGISTER
001026  177400                  RKDS:   177400          ;DRIVE STATUS REGISTER
001030  177416                  DBR:    177416          ;DATA BUFFER REGISTER
001032  177414                  MA:     177414          ;MAINTENANCE REGISTER
001034  000220                  VECTOR: 220
001036  000222                  STATUS: 222
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;RK11 DEDICATE REGISTERS (MEMORY)
                                ;
001040  000000                  FLAG:   0               ;INTERNAL PROGRAM FLAG
001042  146723                  RANNU:  146723          ;RANDOM NUMBER PRIME
001044  000000                  WRDCT:  0               ;WRDING WORD COUNT
001046  000000                  TRACK:  0               ;WRDING DAE
001050  000000                  DMA:    0               ;WRDING DAR
001052  000000                  PATNU:  0               ;DATA PATTERN INDEX
001054  000000                  BUF:    0               ;WRDING DATA BUFFER (OUT-IN)
001056  000000                  TWRDCT: 0               ;TEMP WORD COUNT
001060  000000                  TDMA:   0               ;TEMP DAR
001062  000000                  SWRDCT: 0               ;STANDARD WORD COUNT
001064  000000                  ERCOUNT:0               ;ERROR COUNT FOR MESSAGES.
001066  000000                  SAVE:   0
001070  000000                  SAV1:   0
001072  000000                  PASS:   0
                                ;
                                ;RF11 WRD REGISTERS
                                ;(CAN BE CHANGED IN ANY ROUTINE)
001074  000000                  WRD:    0
001076  000000                  WRD1:   0
001100  000000                  WRD2:   0
```

```
001102  000005                  START:  RESET                   ;CLEAR THE WORLD
001104  012706  001000                  MOV     #1000,%6        ;SET UP STACK
001110  012767  000006  176666          MOV     #6,4
001116  005067  176664                  CLR     6
001122  012767  000340  177056          MOV     #340,2@6        ;LOCK UP INTERRUPTS
001130  012767  010564  176672          MOV     #EMTRP,3@       ;SET UP TTY POINTER
001136  012767  000340  176666          MOV     #340,32         ;LOCK UP INTERRUPTS
001144  012767  010106  176662          MOV     #DISK,34        ;SET UP DISK HANDLER POINTER
001152  012767  000340  176656          MOV     #340,36         ;LOCK UP INTERRRUPTS
001160  012777  000340  177614          MOV     #340,@CSR       ;LOCK UP INTERRUPT LEVELS
001166  012767  010036  176624          MOV     #LOOP,2@        ;SET UP FOR SCOPE LOOP
001174  012767  000340  176620          MOV     #340,22         ;LOCK UP PRIORITY
001202  005067  177640                  CLR     TRACK           ;CLEAR TRACK REGISTERS
001206  005067  177636                  CLR     DMA             ;CLEAR DAR REGISTERS
```

```
                                ;WE ARE NOW ENTERING THE STATIC TEST
                                ;IF THE OPERATOR WOULD LIKE TO CHECK
                                ;THE DISK REGISTERS PRIOR TO ENTERING THIS
                                ;TEST WE HAVE SOME HANDY ROUTINES
                                ;WHICH WOULD ALLOW YOU TO LOAD THESE
                                ;REGISTERS UNDER SWITCH REGISTER CONTROL
                                ;PLEASE REFERENCE THE STARTING ADDRESS
                                ;TO SEE WHICH ROUTINE BEST SUITS YOUR
                                ;PROBLEM.
                                ;
                                ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
                                ;TO CLEAR ALL THE DISK REGISTERS
                                ;TEST CONTROL REGISTER
001212  000005                  STAI1:  RESET                   ;CLEAR THE WORLD
001214  017767  177574  177652          MOV     @DCS,WRD        ;FETCH CONTROL REGISTER
001222  032767  177577  177644          BIT     #177577,WRD     ;IS IT CLEARED
001230  001405                          BEQ     XSTAI1          ;REGISTER OK (TEST READY)
                                        ERROR   \N
001232  012767  000000  177624  ERR0:   MOV     #0,ERCOUNT      ;SET UP ERROR COUNT 0
        000001                          N=N+1
001240  004567  007160                  JSR     %5,STAER        ;REPORT STATIC ERROR
001244  105767  177624          XSTAI1: TSTB    WRD             ;TEST FOR READY
001250  001005                          BNE     LPST1           ;BRANCH IF READY
                                        ERROR   \N
001252  012767  000001  177604  ERR1:   MOV     #1,ERCOUNT      ;SET UP ERROR COUNT 1
        000002                          N=N+1
001260  004567  007140                  JSR     %5,STAER        ;REPORT READY NOT SET
001264  000004                  LPST1:  SCOPE                   ;ENTER SCOPE LOOP
001266  001212                          STAI1
                                ;TEST CURRENT ADDRESS REGISTER
                                ;
001270  000005                  STAI2:  RESET                   ;CLEAR THE WORLD
001272  017767  177522  177574          MOV     @CMA,WRD        ;FETCH CMA REGISTER
001300  001405                          BEQ     LPST2           ;REGISTER OK
                                        ERROR   \N
001302  012767  000002  177554  ERR2:   MOV     #2,ERCOUNT      ;SET UP ERROR COUNT 2
        000003                          N=N+1
001310  004567  007110                  JSR     %5,STAER        ;REPORT ERROR
001314  000004                  LPST2:  SCOPE                   ;ENTER SCOPE LOOP
001316  001270                          STAI2
                                ;
                                ;TEST WORD COUNT REGISTER
                                ;
001320  000005                  STAI3:  RESET                   ;CLEAR THE WORLD
001322  017767  177470  177544          MOV     @WC,WRD ;FETCH WC REG.
001330  001405                          BEQ     LPST3           ;REGISTER OK
                                        ERROR   \N
001332  012767  000003  177524  ERR3:   MOV     #3,ERCOUNT      ;SET UP ERROR COUNT 3
        000004                          N=N+1
001340  004567  007060                  JSR     %5,STAER        ;REPORT ERROR
001344  000004                  LPST3:  SCOPE                   ;ENTER SCOPE LOOP
001346  001320                          STAI3
```

```
                                ;
                                ;TEST DISK ADDRESS REGISTER
                                ;
001350  000005                  STAI4:  RESET                   ;CLEAR THE WORLD
001352  017767  177444  177514          MOV     @DAR,WRD        ;FETCH DAR REGISTER
001360  001405                          BEQ     LPST4           ;REGISTER CLEARED
                                        ERROR   \N
001362  012767  000004  177474  ERR4:   MOV     #4,ERCOUNT      ;SET UP ERROR COUNT 4
        000005                          N=N+1
001370  004567  007030                  JSR     %5,STAER        ;REPORT ERROR
001374  000004                  LPST4:  SCOPE                   ;ENTER SCOPE LOOP
001376  001350                          STAI4
                                ;
                                ;TEST CONTROL ERROR REGISTER
                                ;
001400  000005                  STAI5:  RESET                   ;CLEAR THE WORLD
001402  017767  177416  177464          MOV     @RKER,WRD       ;FETCH RKER REGISTERS
001410  001405                          BEQ     LPST5           ;RKER CLEARED
                                        ERROR   \N
001412  012767  000005  177444  ERR5:   MOV     #5,ERCOUNT      ;SET UP ERROR COUNT 5
        000006                          N=N+1
001420  004567  007000                  JSR     %5,STAER        ;REPORT ERROR
001424  000004                  LPST5:  SCOPE                   ;ENTER SCOPE LOOP
001426  001400                          STAI5
                                ;
                                ;TEST DATA BUFFER REGISTER
                                ;
001430  000005                  STAI6:  RESET                   ;CLEAR THE WORLD
001432  017767  177372  177434          MOV     @DBR,WRD        ;FETCH DBR REGISTER
001440  001405                          BEQ     LPST6           ;DBR CLEARED
                                        ERROR   \N
001442  012767  000006  177414  ERR6:   MOV     #6,ERCOUNT      ;SET UP ERROR COUNT 6
        000007                          N=N+1
001450  004567  006750                  JSR     %5,STAER        ;REPORT ERROR
001454  000004                  LPST6:  SCOPE                   ;ENTER SCOPE LOOP
001456  001430                          STAI6
                                ;
                                ;TEST MAINTENANCE REGISTER
                                ;
001460  000005                  STAI7:  RESET                   ;CLEAR THE WORLD
001462  017767  177344  177404          MOV     @MA,WRD ;FETCH MAINTENANCE REG
001470  001405                          BEQ     LPST7           ;MA CLEARED
                                        ERROR   \N
001472  012767  000007  177364  ERR7:   MOV     #7,ERCOUNT      ;SET UP ERROR COUNT 7
        000010                          N=N+1
001500  004567  006720                  JSR     %5,STAER        ;REPORT ERROR
001504  000004                  LPST7:  SCOPE                   ;ENTER SCOPE LOOP
001506  001460                          STAI7
                                ;
                                ;CAN WE SET THE FUNCTION BITS IN THE DCS REG,
                                ;BITS 7,6,5,4,3,2&1
                                ;
001510  012777  000340  177264  STAI10: MOV     #340,@CSR
001516  112777  000176  177270          MOVB    #176,@DCS       ;SET DISK FUNCTION BITS
```

```
001524  017767  177264  177342          MOV     @DCS,WRD        ;FETCH FUNCTION BITS
001532  022767  000376  177334          CMP     #376,WRD        ;ARE THE FUNCTION BITS SET
001540  001410                          BEQ     LPST10          ;FUNCTION BITS SET
001542  012767  000376  177326          MOV     #376,WRD1       ;SET UP FOR ERROR REPORT
                                        ERROR   \N
001550  012767  000010  177306  ERR10:  MOV     #10,ERCOUNT     ;SET UP ERROR COUNT 10
        000011                          N=N+1
001556  004567  006714                  JSR     %5,STAER1       ;REPORT ERROR (ERROR IN FUNCTION BITS)
001562  000004                  LPST10: SCOPE                   ;ENTER SCOPE LOOP
001564  001510                          STA110
```

```
                                        ;
                                        ;WILL DISK CLEAR CLEAR THE FUNCTION BITS
                                        ;
001566  012777  000340  177206  STAI11: MOV     #340,@CSR
001574  012777  000176  177212          MOV     #176,@DCS       ;SET DISK FUNCTION BITS
001602  012777  000001  177204          MOV     #BIT0,@DCS      ;EXECUTE DISK CLEAR
001610  105777  177200                  TSTB    @DCS
001614  100375                          BPL     .-4
001616  017767  177172  177250          MOV     @DCS,WRD        ;FETCH CONTROL REG
001624  022767  000200  177242          CMP     #200,WRD        ;IS ONLY READY SET
001632  001405                          BEQ     LPST11          ;REGISTER CLEARED
                                        ERROR   \N
001634  012767  000011  177222  ERR11:  MOV     #11,ERCOUNT     ;SET UP ERROR COUNT 11
        000012                          N=N+1
001642  004567  006556                  JSR     %5,STAER        ;REPORT ERROR
001646  000004                  LPST11: SCOPE                   ;ENTER SCOPE LOOP
001650  001566                          STAI11
                                        ;
                                        ;CAN WE SET ALL THE CMA BITS
                                        ;
001652  012767  177777  177216  STAI12: MOV     #177777,WRD1    ;SET UP CURRENT ADDR. OF ALL ONES
001660  016777  177212  177132          MOV     WRD1,@CMA       ;LOAD CMA
001666  017767  177126  177200          MOV     @CMA,WRD        ;FETCH CMA
001674  026767  177176  177172          CMP     WRD1,WRD        ;COMPARE FOR ALL BITS SET
001702  001405                          BEQ     LPST12          ;ALL BITS SET?
                                        ERROR   \N
001704  012767  000012  177152  ERR12:  MOV     #12,ERCOUNT     ;SET UP ERROR COUNT 12
        000013                          N=N+1
001712  004567  006560                  JSR     %5,STAER1       ;REPORT ERROR
001716  000004                  LPST12: SCOPE                   ;ENTER SCOPE LOOP
001720  001652                          STAI12
                                        ;
                                        ;WILL DISK CLEAR - CLEAR THE CMA REGISTER
                                        ;
001722  012777  177777  177070  STAI13: MOV     #177777,@CMA    ;SET CMA EQUAL TO ALL ONES
001730  012777  000001  177056          MOV     #BIT0,@DCS      ;EXECUTE DISK CLEAR
001736  105777  177052                  TSTB    @DCS
001742  100375                          BPL     .-4
001744  005777  177050                  TST     @CMA            ;TEST FOR BIT0 SET IN CMA (READ ONLY BIT)
001750  001410                          BEQ     LPST13          ;WAS THE REST OF CMA CLEARED?
001752  017767  177036  177114          MOV     @DCS,WRD        ;NO! FETCH CMA REG.
                                        ERROR   \N
001760  012767  000013  177076  ERR13:  MOV     #13,ERCOUNT     ;SET UP ERROR COUNT 13
        000014                          N=N+1
001766  004567  006432                  JSR     %5,STAER        ;REPORT ERROR
001772  000004                  LPST13: SCOPE                   ;ENTER SCOPE LOOP
001774  001722                          STAI13
                                        ;
                                        ;
                                        ;CAN WE SET ALL WORD COUNT BITS
                                        ;
001776  012767  177777  177072  STAI14: MOV     #177777,WRD1
002004  016777  177066  177004          MOV     WRD1,@WC        ;SET WORD COUNT EQUAL TO ALL ONES
002012  017767  177000  177054          MOV     @WC,WRD ;FETCH WORD COUNT
```

```
002020  026767  177052  177046          CMP     WRD1,WRD        ;IS WORD COUNT EQUAL TO ALL ONES
002026  001405                          BEQ     LPST14          ;WORD COUNT EQUAL TO ALL ONES
                                        ERROR   \N
002030  012767  000014  177026  ERR14:  MOV     #14,ERCOUNT     ;SET UP ERROR COUNT 14
        000015                          N=N+1
002036  004567  006434                  JSR     %5,STAER1       ;REPORT WORD COUNT NOT EQUAL TO ALL ONES
002042  000004                  LPST14: SCOPE
002044  001776                          STAI14
```

```
                                ;
                                ;WILL DISK CLEAR=CLEAR THE WORD COUNT REGISTER
                                ;
002046  012777  177777  176742  STAI15: MOV     #177777,@WC     ;SET WC REGISTER EQUAL TO ALL ONES
002054  012777  000001  176732          MOV     #BIT0,@DCS      ;EXECUTE DISK CLEAR
002062  105777  176726                  TSTB    @DCS
002066  100375                          BPL     .-4
002070  017767  176722  176776          MOV     @WC,WRD ;FETCH WORD COUNT REGISTER
002076  001405                          BEQ     LPST15          ;YES! EXIT
                                        ERROR   \N
002100  012767  000015  176756  ERR15:  MOV     #15,ERCOUNT     ;SET UP ERROR COUNT 15
        000016                          N=N+1
002106  004567  006312                  JSR     %5,STAER        ;REPORT ERROR
002112  000004                  LPST15: SCOPE                   ;ENTER SCOPE LOOP
002114  002046                          STAI15
                                ;
                                ;CAN WE SET ALL THE BITS IN THE DAR REGISTER,
                                ;
002116  012767  177777  176752  STAI16: MOV     #177777,WRD1
002124  016777  176746  176670          MOV     WRD1,@DAR       ;SET DAR TO ALL ONES
002132  017767  176664  176734          MOV     @DAR,WRD        ;FETCH DAR REGISTER
002140  026767  176732  176726          CMP     WRD1,WRD        ;ARE ALL BITS SET
002146  001405                          BEQ     LPST16          ;YES! EXIT
                                        ERROR   \N
002150  012767  000016  176706  ERR16:  MOV     #16,ERCOUNT     ;SET UP ERROR COUNT 16
        000017                          N=N+1
002156  004567  006314                  JSR     %5,STAER1       ;NOT ALL BITS SET REPORT ERROR
002162  000004                  LPST16: SCOPE                   ;ENTER SCOPE LOOP
002164  002116                          STAI16
                                ;
                                ;CAN WE CLEAR THE DAR REG. WITH DISK CLEAR,
                                ;
002166  012777  177777  176626  STAI17: MOV     #177777,@DAR    ;SET DAR TO ALL ONES
002174  012777  000001  176612          MOV     #BIT0,@DCS      ;EXECUTE DISK CLEAR
002202  105777  176606                  TSTB    @DCS
002206  100375                          BPL     .-4
002210  005777  176606                  TST     @DAR            ;TEST FOR ZERO DAR
002214  001410                          BEQ     LPST17          ;YES EXIT
002216  017767  176600  176650          MOV     @DAR,WRD        ;NO BITS SET IN DAR
                                        ERROR   \N
002224  012767  000017  176632  ERR17:  MOV     #17,ERCOUNT     ;SET UP ERROR COUNT 17
        000020                          N=N+1
002232  004567  006166                  JSR     %5,STAER        ;REPORT ERROR
002236  000004                  LPST17: SCOPE                   ;ENTER SCOPE LOOP
002240  002166                          STAI17
                                ;
```

```
                                        ;
                                        ;DO ONE WORD WRITE FOLLOWED BY ONE WORD WRITE CONTINUE
                                        ;ERROR1=BUSY FAILED TO SET
                                        ;ERROR2=BUSY FAILED TO CLEAR
                                        ;ERROR3=CONTROL ERROR WHEN X-FERING DATA
                                        ;ERROR5=DAR FAILED TO INCREMENT WHEN X-FERING DATA
                                        ;ERROR6=WORD COUNT FAILED TO OVERFLOW
                                        ;ERROR7=CMA FAILED TO INCREMENT
                                        ;
                                        ;
                                        ;* * * EXECUTE THE ONE WORD WRITE* * *
002242  012777  000001  176544  STAI22: MOV     #BIT0,@DCS      ;CLEAR THE DISK WORLD
002250  105777  176540                  TSTB    @DCS
002254  100375                          BPL     .-4
002256  012767  177777  000660          MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
002264  012777  011144  176526          MOV     #OUTBUF,@CMA    ;SET UP CURRENT ADDRESS
002272  012777  177777  176516          MOV     #-1,@WC         ;SET WORD COUNT TO -1
002300  052777  000003  176506          BIS     #3,@DCS         ;GO WRITE
002306  105777  176502                  TSTB    @DCS            ;TEST FOR RDY=0
002312  100011                          BPL     STAI23          ;RDY=0
002314  017767  176474  176552          MOV     @DCS,WRD        ;BUSY NOT SET; FETCH DCS
                                ERROR1: ERROR   \N
002322  012767  000020  176534  ERR20:  MOV     #20,ERCOUNT     ;SET UP ERROR COUNT 20
        000021                          N=N+1
002330  004567  006070                  JSR     %5,STAER        ;REPORT ERROR
002334  000742                          BR      STAI22          ;RESTART TEST
002336  005067  176532          STAI23: CLR     WRD
002342  005267  176526          INCWAT: INC     WRD             ;WAIT FOR BUSY=0
002346  105777  176442                  TSTB    @DCS            ;IS BUSY CLEARED
002352  100417                          BMI     STAI24          ;FLAG CLEARED
002354  005767  176514                  TST     WRD             ;HAVE WE WAITED LONG ENOUGH
002360  001370                          BNE     INCWAT          ;RDY FAILED TO SET
002362  017767  176426  176504          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS REG
002370  052767  000002  176500          BIS     #2,WRD1 ;WANT DCS SHOULD CONTAIN
                                ERROR2: ERROR   \N
002376  012767  000021  176460  ERR21:  MOV     #21,ERCOUNT     ;SET UP ERROR COUNT 21
        000022                          N=N+1
002404  004567  006066                  JSR     %5,STAER1       ;REPORT ERROR
002410  000714                          BR      STAI22          ;RESTART TEST
002412  017767  176376  176454  STAI24: MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS REG
002420  005767  176450                  TST     WRD             ;IS ERROR FLAG SET
002424  100012                          BPL     STAI26          ;NO; X-FER OK
002426  012767  000202  176442          MOV     #202,WRD1       ;WHAT DCS SHOULD CONTAIN
                                ERROR3: ERROR   \N
002434  012767  000022  176422  ERR22:  MOV     #22,ERCOUNT     ;SET UP ERROR COUNT 22
        000023                          N=N+1
002442  004567  006030                  JSR     %5,STAER1       ;REPORT ERROR
002446  000167  177570                  JMP     STAI22          ;RESTART TEST
002452  017767  176344  176414  STAI26: MOV     @DAR,WRD        ;WAS DAR INCREMENTED BY 1
002460  022767  000001  176406          CMP     #BIT0,WRD       ;IS DAR CORRECT
002466  001412                          BEQ     STAI27          ;DAR OK
002470  012767  000001  176400          MOV     #BIT0,WRD1      ;WHAT DAR SHOULD CONTAIN
                                ERROR5: ERROR   \N
002476  012767  000023  176360  ERR23:  MOV     #23,ERCOUNT     ;SET UP ERROR COUNT 23
```

```
        000024                          N=N+1
002504  004567  005766                  JSR     X5,STAER1       ;REPORT ERROR
002510  000167  177526                  JMP     STAI22          ;RESTART TEST
```

```
002514  017767  176276  176352  STAI27: MOV     @WC,WRD ;FETCH WORD COUNT
002522  001407                          BEQ     STAI30          ;WORD COUNT DID OVERFLOW
                                ERROR6: ERROR   \N
002524  012767  000024  176332  ERR24:  MOV     #24,ERCOUNT     ;SET UP ERROR COUNT 24
        000025                          N=N+1
002532  004567  005666                  JSR     %5,STAER        ;WORD COUNT FAILED TO OVERFLOW
002536  000167  177500                  JMP     STAI22          ;RESTART TEST
002542  017767  176252  176324  STAI30: MOV     @CMA,WRD        ;FETCH CURRENT ADDRESS
002550  012767  011146  176320          MOV     #OUTBUF+2,WRD1  ;WHAT CMA SHOULD EQUAL
002556  026767  176314  176310          CMP     WRD1,WRD        ;IS CMA CORRECT
002564  001407                          BEQ     STAI31          ;YES EXECUTE CONTINUE
                                ERROR7: ERROR   \N
002566  012767  000025  176270  ERR25:  MOV     #25,ERCOUNT     ;SET UP ERROR COUNT 25
        000026                          N=N+1
002574  004567  005676                  JSR     %5,STAER1       ;REPORT ERROR IN CMA
002600  000167  177436                  JMP     STAI22          ;ERROR RESTART TEST
```

```
                                ;TAPE 2
                                ;       TEST WRITE
                                ;A WRITE CONTINUE WILL BE EXECUTED NOW
                                ;WORD COUNT WILL BE SET TO -1 AGAIN
                                ;
                                ;EROR10=BUSY NOT SET BY GO
                                ;EROR11=BUSY NOT CLEARED BY OVERFLOW
                                ;EROR12=DISK ERROR OCCURED WHILE X-FERING
                                ;EROR14=DAR DID NOT EQUAL 2 AFTER SECOND X-FER
                                ;EROR15=WC DID NOT OVERFLOW AT THE END OF X-FER
                                ;EROR16=CMA DID NOT EQUAL OUTBUF+2 AT END OF X-FER
                                ;
                                ;
002604  012777  177777  176204  STAI31: MOV     #177777,@WC     ;SET WC TO -1
002612  052777  000001  176174          BIS     #BIT0,@DCS      ;SET GO TO CONTINUE
002620  105777  176170          STAI32: TSTB    @DCS            ;TEST FOR RDY SET
002624  100012                          BPL     STAI32          ;BUSY WAS SET BY GO
002626  017767  176162  176240          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                EROR10: ERROR   \N
002634  012767  000026  176222  ERR26:  MOV     #26,ERCOUNT     ;SET UP ERROR COUNT 26
        000027                          N=N+1
002642  004567  005556                  JSR     %5,STAER        ;REPORT BUSY NOT SET
002646  000167  177370                  JMP     STAI22
002652  005067  176216          STAI32: CLR     WRD
002656  105777  176132          INCBUSY:TSTB    @DCS            ;TST FOR RDY SET BY OVERFLO
002662  100415                          BMI     STAI33          ;READY SET CONTINUE
002664  005267  176204                  INC     WRD             ;WAIT FOR RDY=1
002670  001372                          BNE     INCBUSY         ;GO WAIT FOR RDY
002672  017767  176116  176174          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                EROR11: ERROR   \N
002700  012767  000027  176156  ERR27:  MOV     #27,ERCOUNT     ;SET UP ERROR COUNT 27
        000030                          N=N+1
002706  004567  005512                  JSR     %5,STAER        ;REPORT BUSY NOT CLEARED
002712  000167  177324                  JMP     STAI22          ;RESTART ROUTINE
002716  005777  176072          STAI33: TST     @DCS            ;DID AN ERROR OCCUR WHILE X-FERING
002722  100015                          BPL     STAI35          ;NO CONTINUE
002724  017767  176064  176142          MOV     @DCS,WRD        ;YES; CONTENTS OF DCS
002732  017767  176066  176136          MOV     @RKER,WRD1      ;EXT ERROR BITS
                                EROR12: ERROR   \N
002740  012767  000030  176116  ERR30:  MOV     #30,ERCOUNT     ;SET UP ERROR COUNT 30
        000031                          N=N+1
002746  004567  005524                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED.
002752  000167  177264                  JMP     STAI22          ;RESTART ROUTINE
002756  017767  176040  176110  STAI35: MOV     @DAR,WRD        ;DID DAR INCREMENT ON CONTINUE
002764  012767  000002  176104          MOV     #2,WRD1 ;WHAT DAR SHOULD CONTAIN
002772  026767  176100  176074          CMP     WRD1,WRD        ;IS DAR CORRECT
003000  001407                          BEQ     STAI36          ;DAR OK
                                EROR14: ERROR   \N
003002  012767  000031  176054  ERR31:  MOV     #31,ERCOUNT     ;SET UP ERROR COUNT 31
        000032                          N=N+1
003010  004567  005462                  JSR     %5,STAER1       ;REPORT DAR INCORRECT
003014  000167  177222                  JMP     STAI22          ;RE-START ROUTINE
003020  017767  175772  176046  STAI36: MOV     @WC,WRD ;FETCH WORD COUNT
003026  001407                          BEQ     STAI37          ;WORD COUNT OVERFLOWED
```

```
                        ERROR15: ERROR   \N
003030  012767  000032  176026  ERR32:  MOV     #32,ERCOUNT     ;SET UP ERROR COUNT 32
        000033                  N=N+1
003036  004567  005362                  JSR     %5,STAER        ;REPORT WORD COUNT FAILED TO CLEAR
003042  000167  177174                  JMP     STAI22          ;RESTART ROUTINE
003046  017767  175746  176020  STAI37: MOV     @CMA,WRD        ;FETCH CMA
003054  012767  011150  176014          MOV     #OUTBUF+4,WRD1  ;WHAT CMA SHOULD EQUAL
003062  026767  176010  176004          CMP     WRD1,WRD        ;IS CMA CORRECT
003070  001407                          BEQ     LPST22          ;CMA WAS CORRECT
                        ERROR16: ERROR   \N
003072  012767  000033  175764  ERR33:  MOV     #33,ERCOUNT     ;SET UP ERROR COUNT 33
        000034                  N=N+1
003100  004567  005372                  JSR     %5,STAER1       ;REPORT CMA INCORRECT
003104  000167  177132                  JMP     STAI22          ;RESTART ROUTINE
003110  000004                  LPST22: SCOPE                   ;ENTER SCOPE LOOP
003112  002242                          STAI22
```

```
                        ;               TEST WRITE CHECK
                        ;DO A ONE WORD WRITE CHECK FOLLOWED BY
                        ;A ONE WORD WRITE CHECK CONTINUE
                        ;
                        ;EROR17=BUSY FAILED TO SET
                        ;EROR20=BUSY FAILED TO CLEAR
                        ;EROR21=DISK ERROR WHEN X-FERING DATA
                        ;EROR23=DAR FAILED TO INCREMENT WHEN X-FERING DATA
                        ;EROR24=WORD COUNT FAILED TO OVERFLOW
                        ;EROR25=CMA FAILED TO INCREMENT
                        ;
                        ;
                        ;* * *EXECUTE THE ONE WORD WRITE CHECK* * *
003114  012777  000001  175672  STAI40: MOV     #BIT0,@DCS      ;CLEAR THE DISK WORLD
003122  105777  175666          STAI40: TSTB    @DCS
003126  100375                  STAI40: BPL     .-4
003130  012767  177777  006006  STAI40: MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
003136  012777  011144  175654  STAI40: MOV     #OUTBUF,@CMA    ;SET UP CURRENT ADDRESS
003144  012777  177777  175644  STAI40: MOV     #-1,@WC         ;SET WORD COUNT TO -1
003152  052777  000007  175634  STAI40: BIS     #7,@DCS         ;GO WRITE CHECK
003160  105777  175630          STAI40: TSTB    @DCS            ;TEST FOR READY
003164  100011                  STAI40: BPL     STAI41          ;NOT READY
003166  017767  175622  175700  STAI40: MOV     @DCS,WRD        ;SET: FETCH DCS
                        EROR17: ERROR   \N
003174  012767  000034  175662  ERR34:  MOV     #34,ERCOUNT     ;SET UP ERROR COUNT 34
        000035                  ERR34:  N=N+1
003202  004567  005216          ERR34:  JSR     %5,STAER        ;REPORT ERROR
003206  000742                  ERR34:  BR      STAI40          ;RESTART TEST
003210  005067  175660          STAI41: CLR     WRD
003214  005267  175654          WCWAT:  INC     WRD             ;WAIT FOR READY
003220  105777  175570          WCWAT:  TSTB    @DCS
003224  100417                  WCWAT:  BMI     STAI42          ;FLAG SET
003226  005767  175642          WCWAT:  TST     WRD             ;HAVE WE WAITED LONG ENOUGH
003232  001370                  WCWAT:  BNE     WCWAT           ;READY FAILED TO SET
003234  017767  175554  175632  STAI42: MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS REG
003242  052767  000006  175626  STAI42: BIS     #6,WRD1 ;WANT DCS SHOULD CONTAIN
                        EROR20: ERROR   \N
003250  012767  000035  175606  ERR35:  MOV     #35,ERCOUNT     ;SET UP ERROR COUNT 35
        000036                  ERR35:  N=N+1
003256  004567  005214          ERR35:  JSR     %5,STAER1       ;REPORT ERROR
003262  000714                  ERR35:  BR      STAI40          ;RESTART TEST
003264  017767  175524  175602  STAI42: MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS REG
003272  100012                  STAI42: BPL     STAI44          ;NO: X-FER OK
003274  012767  000206  175574  STAI42: MOV     #206,WRD1
                        EROR21: ERROR   \N
003302  012767  000036  175554  ERR36:  MOV     #36,ERCOUNT     ;SET UP ERROR COUNT 36
        000037                  ERR36:  N=N+1
003310  004567  005162          ERR36:  JSR     %5,STAER1       ;REPORT ERROR
003314  000167  177574          ERR36:  JMP     STAI40          ;RESTART TEST
003320  017767  175476  175546  STAI44: MOV     @DAR,WRD        ;WAS DAR INCREMENTED BY 1
003326  022767  000001  175540  STAI44: CMP     #BIT0,WRD       ;IS DAR CORRECT
003334  001412                  STAI44: BEQ     STAI45          ;DAR OK
003336  012767  000001  175532  STAI44: MOV     #BIT0,WRD1      ;WHAT DAR SHOULD CONTAIN
                        EROR23: ERROR   \N
```

```
003344  012767  000037  175512  ERR37:  MOV     #37,ERCOUNT     ;SET UP ERROR COUNT 37
        000040                  N=N+1
003352  004567  005120                  JSR     %5,STAER1       ;REPORT ERROR
003356  000167  177532                  JMP     STAI40          ;RESTART TEST
003362  017767  175430  175504  STAI45: MOV     @WC,WRD ;FETCH WORD COUNT
003370  001407                  BEQ     STAI46          ;WORD COUNT DID OVERFLOW
                        EROR24: ERROR   \N
003372  012767  000040  175464  ERR40:  MOV     #40,ERCOUNT     ;SET UP ERROR COUNT 40
        000041                  N=N+1
003400  004567  005020                  JSR     %5,STAER        ;WORD COUNT FAILED TO OVERFLOW
003404  000167  177504                  JMP     STAI40          ;RESTART TEST
003410  017767  175404  175456  STAI46: MOV     @CMA,WRD        ;FETCH CURRENT ADDRESS
003416  012767  011146  175452          MOV     #OUTBUF+2,WRD1  ;WHAT CMA SHOULD EQUAL
003424  026767  175446  175442          CMP     WRD1,WRD        ;IS CMA CORRECT
003432  001407                  BEQ     STAI47          ;YES EXECUTE CONTINUE
                        EROR25: ERROR   \N
003434  012767  000041  175422  ERR41:  MOV     #41,ERCOUNT     ;SET UP ERROR COUNT 41
        000042                  N=N+1
003442  004567  005030                  JSR     %5,STAER1       ;REPORT ERROR IN CMA
003446  000167  177442                  JMP     STAI40          ;ERROR RESTART TEST
```

```
                                ;
                                ;A WRITE CHECK CONTINUE CONTINUE WILL BE EXECUTED NOW
                                ;WORD COUNT WILL BE SET TO -1 AGAIN
                                ;
                                ;EROR26=BUSY WAS NOT SET BY GO
                                ;EROR27=BUSY NOT CLEARED BY OVERFLOW
                                ;EROR30=DISK ERROR OCCURRED WHILE X-FERING
                                ;EROR32=DAR DID NOT EQUAL 2 AFTER SECOND X-FER
                                ;EROR33=WC DID NOT=0 AFTER X-FER
                                ;EROR34=CMA DID NOT=OUTBUF+2 AT END OF X-FER
                                ;
                                ;
003452  012777  177777  175336  STAI47: MOV     #177777,@WC     ;SET WC TO -1
003460  052777  000001  175326          BIS     #BIT0,@DCS      ;SET GO TO CONTINUE
003466  105777  175322          STAI50: TSTB    @DCS            ;TEST FOR READY
003472  100012                          BPL     STAI50          ;
003474  017767  175314  175372          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                EROR26: ERROR   \N
003502  012767  000042  175354  ERR42:  MOV     #42,ERCOUNT     ;SET UP ERROR COUNT 42
        000043                          N=N+1
003510  004567  004710                  JSR     %5,STAER        ;REPORT BUSY NOT SET
003514  000167  177374                  JMP     STAI40
003520  005067  175350          STAI50: CLR     WRD
003524  105777  175264          WCBUSI: TSTB    @DCS            ;TST FOR RDY SET BY OVERFLOW
003530  100415                          BMI     STAI51          ;BRANCH IF READY SET
003532  005267  175336                  INC     WRD             ;WAIT FOR BUSY=0
003536  001372                          BNE     WCBUSI          ;GO WAIT FOR RDY
003540  017767  175250  175326          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                EROR27: ERROR   \N
003546  012767  000043  175310  ERR43:  MOV     #43,ERCOUNT     ;SET UP ERROR COUNT 43
        000044                          N=N+1
003554  004567  004644                  JSR     %5,STAER        ;REPORT RDY NOT SET
003560  000167  177330                  JMP     STAI40          ;RESTART ROUTINE
003564  005777  175224          STAI51: TST     @DCS            ;DID AN ERROR OCCUR WHILE X-FERING
003570  100015                          BPL     STAI53          ;NO CONTINUE
003572  017767  175216  175274          MOV     @DCS,WRD        ;YES! CONTENTS OF DCS
003600  012767  000206  175270          MOV     #206,WRD1
                                EROR30: ERROR   \N
003606  012767  000044  175250  ERR44:  MOV     #44,ERCOUNT     ;SET UP ERROR COUNT 44
        000045                          N=N+1
003614  004567  004656                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED
003620  000167  177270                  JMP     STAI40          ;RESTART ROUTINE
003624  017767  175172  175242  STAI53: MOV     @DAR,WRD        ;DID DAR INCREMENT ON CONTINUE
003632  012767  000002  175236          MOV     #2,WRD1 ;WHAT DAR SHOULD CONTAIN
003640  026767  175232  175226          CMP     WRD1,WRD        ;IS DAR CORRECT
003646  001407                          BEQ     STAI54          ;DAR OK
                                EROR32: ERROR   \N
003650  012767  000045  175206  ERR45:  MOV     #45,ERCOUNT     ;SET UP ERROR COUNT 45
        000046                          N=N+1
003656  004567  004614                  JSR     %5,STAER1       ;REPORT DAR INCORRECT
003662  000167  177226                  JMP     STAI40          ;RE-START ROUTINE
003666  017767  175124  175200  STAI54: MOV     @WC,WRD ;FETCH WORD COUNT
003674  001407                          BEQ     STAI55          ;WORD COUNT OVERFLOWED
                                EROR33: ERROR   \N
```

```
003676  012767  000046  175160  ERR46:  MOV     #46,ERCOUNT     ;SET UP ERROR COUNT 46
        000047                  N=N+1
003704  004567  004514                  JSR     %5,STAER        ;REPORT WORD COUNT FAILED TO CLEAR
003710  000167  177200                  JMP     STAI40          ;RESTART ROUTINE
003714  017767  175100  175152  STAI55: MOV     @CMA,WRD        ;FETCH CMA
003722  012767  011150  175146          MOV     #OUTBUF+4,WRD1  ;WHAT CMA CHOULD EQUAL
003730  026767  175142  175136          CMP     WRD1,WRD        ;IS CMA CORRECT
003736  001407                          BEQ     LPST40          ;CMA WAS CORRECT
                                EROR34: ERROR   \N
003740  012767  000047  175116  ERR47:  MOV     #47,ERCOUNT     ;SET UP ERROR COUNT 47
        000050                  N=N+1
003746  004567  004524                  JSR     %5,STAER1       ;REPORT CMA INCORRECT
003752  000167  177136                  JMP     STAI40          ;RESTART ROUTINE
003756  000004                  LPST40: SCOPE                   ;ENTER SCOPE LOOP
003760  003114                          STAI40
```

```
                                ;       TEST READ
                                ;ONE WORD READ FOLLOWED BY A ONE WORD
                                ;READ CONTINUE.
                                ;
                                ;ERROR35=BUSY WAS NOT SET BY GO
                                ;ERROR36=BUSY NOT CLEARED BY OVERFLOW
                                ;ERROR37=DISK ERROR OCCURRED WHILE X-FERING DATA
                                ;ERROR41=DAR FAILED TO INCREMENT WHEN X-FERING DATA
                                ;ERROR42=WORD COUNT FAILED TO OVERFLOW
                                ;ERROR43=CMA FAILED TO INCREMENT
                                ;
                                ;
                                ;* * *EXECUTE THE ONE WORD READ* * *
003762  012777  000001  175024  STAI56: MOV     #BIT0,@DCS      ;CLEAR THE DISK WORLD
003770  105777  175020          STAI56: TSTB    @DCS
003774  100375                  STAI56: BPL     .-4
003776  012767  177777  005140  STAI56: MOV     #177777,OUTBUF  ;DATA TO BE X-FERED
004004  012777  011144  175006  STAI56: MOV     #OUTBUF,@CMA    ;SET UP CURRENT ADDRESS
004012  012777  177777  174776  STAI56: MOV     #-1,@WC         ;SET WORD COUNT TO -1
004020  052777  000005  174766  STAI56: BIS     #5,@DCS         ;GO READ
004026  105777  174762          STAI56: TSTB    @DCS            ;TEST FOR BUSY=1
004032  100011                  STAI56: BPL     STAI57          ;BUSY SET
004034  017767  174754  175032  STAI56: MOV     @DCS,WRD        ;BUSY NOT SET! FETCH DCS
                                ERROR35: ERROR  \N
004042  012767  000050  175014  ERR50:  MOV     #50,ERCOUNT     ;SET UP ERROR COUNT 50
        000051                  ERR50:  N=N+1
004050  004567  004350          ERR50:  JSR     %5,STAER        ;REPORT ERROR
004054  000742                  ERR50:  BR      STAI56          ;RESTART TEST
004056  005067  175012          STAI57: CLR     WRD
004062  005267  175006          RDWAT:  INC     WRD             ;WAIT FOR BUSY=0
004066  105777  174722                  TSTB    @DCS            ;IS BUSY CLEARED
004072  100417                          BMI     STAI60          ;FLAG CLEARED
004074  005767  174774                  TST     WRD             ;HAVE WE WAITED LONG ENOUGH
004100  001370                          BNE     RDWAT           ;BUSY FAILED TO CLEAR
004102  017767  174706  174764          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS REG
004110  052767  000005  174760          BIS     #5,WRD1 ;WANT DCS SHOULD CONTAIN
                                ERROR36: ERROR  \N
004116  012767  000051  174740  ERR51:  MOV     #51,ERCOUNT     ;SET UP ERROR COUNT 51
        000052                  ERR51:  N=N+1
004124  004567  004346          ERR51:  JSR     %5,STAER1       ;REPORT ERROR
004130  000714                  ERR51:  BR      STAI56          ;RESTART TEST
004132  017767  174656  174734  STAI60: MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS REG
004140  005767  174730                  TST     WRD             ;IS ERROR FLAG SET
004144  100012                          BPL     STAI62          ;NO! X-FER OK
004146  012767  000003  174722          MOV     #3,WRD1 ;WHAT DCS SHOULD CONTAIN
                                ERROR37: ERROR  \N
004154  012767  000052  174702  ERR52:  MOV     #52,ERCOUNT     ;SET UP ERROR COUNT 52
        000053                  ERR52:  N=N+1
004162  004567  004310          ERR52:  JSR     %5,STAER1       ;REPORT ERROR
004166  000167  177570          ERR52:  JMP     STAI56          ;RESTART TEST
004172  017767  174624  174674  STAI62: MOV     @DAR,WRD        ;WAS DAR INCREMENTED BY 1
004200  022767  000001  174666          CMP     #BIT0,WRD       ;IS DAR CORRECT
004206  001412                          BEQ     STAI63          ;DAR OK
004210  012767  000001  174660          MOV     #BIT0,WRD1      ;WHAT DAR SHOULD CONTAIN
```

```
                                ERROR41: ERROR   \N
004216  012767  000053  174640  ERR53:  MOV     #53,ERCOUNT     ;SET UP ERROR COUNT 53
        000054                          N=N+1
004224  004567  004246                  JSR     %5,STAER1       ;REPORT ERROR
004230  000167  177526                  JMP     STAI56          ;RESTART TEST
004234  017767  174556  174632  STAI63: MOV     @WC,WRD ;FETCH WORD COUNT
004242  001407                          BEQ     STAI64          ;WORD COUNT DID OVERFLOW
                                ERROR42: ERROR   \N
004244  012767  000054  174612  ERR54:  MOV     #54,ERCOUNT     ;SET UP ERROR COUNT 54
        000055                          N=N+1
004252  004567  004146                  JSR     %5,STAER        ;WORD COUNT FAILED TO OVERFLOW
004256  000167  177500                  JMP     STAI56          ;RESTART TEST
004262  017767  174532  174604  STAI64: MOV     @CMA,WRD        ;FETCH CURRENT ADDRESS
004270  012767  011146  174600          MOV     #OUTBUF+2,WRD1  ;WHAT CMA SHOULD EQUAL
004276  026767  174574  174570          CMP     WRD1,WRD        ;IS CMA CORRECT
004304  001407                          BEQ     STAI65          ;YES EXECUTE CONTINUE
                                ERROR43: ERROR   \N
004306  012767  000055  174550  ERR55:  MOV     #55,ERCOUNT     ;SET UP ERROR COUNT 55
        000056                          N=N+1
004314  004567  004156                  JSR     %5,STAER1       ;REPORT ERROR IN CMA
004320  000167  177436                  JMP     STAI56          ;ERROR RESTART TEST
```

```
                                ;A READ CONTINUE WILL BE EXECUTED NOW
                                ;WORD COUNT WILL BE SET TO -1 AGAIN
                                ;
                                ;EROR44=BUSY NOT SET BY GO
                                ;EBOR45=BUSY NOT CLEARED BY OVERFLOW
                                ;EROR46=DISK ERROR OCCURRED WHILE X-FERING
                                ;EROR50=DAR DID NOT EQUAL 2 AFTER SECOND X-FER
                                ;EROR51=WC FAILED TO OVERFLOW ON READ CONTINUE
                                ;EROR52=CMA DID NOT EQUAL OUTBUF+2 AFTER READ CONTINUE
                                ;
                                ;
004324  012777  177777  174464  STAI65: MOV     #177777,@WC     ;SET WC TO -1
004332  052777  000001  174454          BIS     #BIT0,@DCS      ;SET GO TO CONTINUE
004340  105777  174450                  TSTB    @DCS            ;TEST FOR RDY=0
004344  100012                        BPL     STAI66          ;RDY SET
004346  017767  174442  174520          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                EROR44: ERROR   \N
004354  012767  000056  174502  ERR56:  MOV     #56,ERCOUNT     ;SET UP ERROR COUNT 56
        000057                          N=N+1
004362  004567  004036                  JSR     X5,STAER        ;REPORT BUSY NOT SET
004366  000167  177370                  JMP     STAI56
004372  005067  174476          STAI66: CLR     WRD
004376  105777  174412          INCRD:  TSTB    @DCS            ;TST FOR RDY SET BY OVERFLOW
004402  100415                        BMI     STAI67          ;RDY SET CONTINUE
004404  005267  174464                  INC     WRD             ;WAIT FOR BUSY=0
004410  001372                        BNE     INCRD           ;GO WAIT FOR BUSY
004412  017767  174376  174454          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                EROR45: ERROR   \N
004420  012767  000057  174436  ERR57:  MOV     #57,ERCOUNT     ;SET UP ERROR COUNT 57
        000060                          N=N+1
004426  004567  003772                  JSR     X5,STAER        ;REPORT BUSY NOT CLEARED
004432  000167  177324                  JMP     STAI56          ;RESTART ROUTINE
004436  005777  174352          STAI67: TST     @DCS            ;DID AN ERROR OCCUR WHILE X-FERING
004442  100015                        BPL     STAI71          ;NO CONTINUE
004444  017767  174344  174422          MOV     @DCS,WRD        ;YES! CONTENTS OF DCS
004452  017767  174346  174416          MOV     @RKER,WRD1      ;EXT ERROR BITS
                                EROR46: ERROR   \N
004460  012767  000060  174376  ERR60:  MOV     #60,ERCOUNT     ;SET UP ERROR COUNT 60
        000061                          N=N+1
004466  004567  004004                  JSR     X5,STAER1       ;REPORT ERROR OCCURRED
004472  000167  175544                  JMP     STAI22          ;RESTART ROUTINE
004476  017767  174320  174370  STAI71: MOV     @DAR,WRD        ;DID DAR INCREMENT ON CONTINUE
004504  012767  000002  174364          MOV     #2,WRD1 ;WHAT DAR SHOULD CONTAIN
004512  026767  174360  174354          CMP     WRD1,WRD        ;IS DAR CORRECT
004520  001407                        BEQ     STAI72          ;DAR OK
                                EROR50: ERROR   \N
004522  012767  000061  174334  ERR61:  MOV     #61,ERCOUNT     ;SET UP ERROR COUNT 61
        000062                          N=N+1
004530  004567  003742                  JSR     X5,STAER1       ;REPORT DAR INCORRECT
004534  000167  177222                  JMP     STAI56          ;RESTART ROUTINE
004540  017767  174252  174326  STAI72: MOV     @WC,WRD ;FETCH WORD COUNT
004546  001407                        BEQ     STAI73          ;WORD COUNT OVERFLOWED
                                EROR51: ERROR   \N
004550  012767  000062  174306  ERR62:  MOV     #62,ERCOUNT     ;SET UP ERROR COUNT 62
```

```
        000063                          N=N+1
004556  004567  003642                  JSR     %5,STAER        ;REPORT WORD COUNT FAILED TO CLEAR
004562  000167  177174                  JMP     STAI56          ;RESTART ROUTINE
004566  017767  174226  174300  STAI73: MOV     @CMA,WRD        ;FETCH CMA
004574  012767  011150  174274          MOV     #OUTBUF+4,WRD1  ;WHAT CMA SHOULD EQUAL
004602  026767  174270  174264          CMP     WRD1,WRD        ;IS CMA CORRECT
004610  001407                          BEQ     LPST56          ;CMA WAS CORRECT
                                ERROR52: ERROR  \N
004612  012767  000063  174244  ERR63:  MOV     #63,ERCOUNT     ;SET UP ERROR COUNT 63
        000064                          N=N+1
004620  004567  003652                  JSR     %5,STAER1       ;REPORT CMA INCORRECT
004624  000167  177132                  JMP     STAI56          ;RESTART ROUTINE
004630  000004                  LPST56: SCOPE                   ;ENTER SCOPE LOOP
004632  003762                          STAI56
```

```
                                        ;
                                        ;
                                        ;
                                        ;THIS ROUTINE TESTS THE ABILITY OF THE CONTROL NOT
                                        ;TO INCREMENT THE CURRENT ADDRESS REGISTER WHEN
                                        ;BIT11 OF THE RKCS REGISTER IS SET
                                        ;
                                        ;
004634  012777  000001  174152  INHCMA: MOV     #BIT0,@DCS      ;CLEAR THE CONTROL
004642  105777  174146                  TSTB    @DCS
004646  100375                          BPL     .-4
004650  012777  177700  174140          MOV     #-100,@WC       ;LOAD THE WORD COUNT
004656  012777  011144  174134          MOV     #OUTBUF,@CMA    ;LOAD THE CURRENT ADDRESS
004664  012777  004003  174122          MOV     #4003,@DCS      ;LOAD FUNCTION REGISTER
004672  105777  174116                  TSTB    @DCS
004676  100375                          BPL     .-4
004700  005777  174110                  TST     @DCS            ;TEST FOR ERRORS
004704  100014                          BPL     CMAOK           ;BRANCH IF NO ERRORS
004706  017767  174102  174162          MOV     @DCS,WRD1
004714  017767  174104  174152          MOV     @RKER,WRD
                                        ERROR   \N
004722  012767  000064  174134  ERR64:  MOV     #64,ERCOUNT     ;SET UP ERROR COUNT 64
        000065                          N=N+1
004730  004567  003542                  JSR     %5,STAER1       ;REPORT ERROR OCCURRED
004734  000737                          BR      INHCMA          ;LOOP ON ERROR
004736  012767  011144  174132  CMAOK:  MOV     #OUTBUF,WRD1    ;FETCH CORRECT CURRENT ADDRESS
004744  017767  174050  174122          MOV     @CMA,WRD
004752  026767  174120  174114          CMP     WRD1,WRD        ;DID THE CUURENT ADRESS INCREMENT
004760  001406                          BEQ     LPINH           ;BRANCH CMA DID NOT INCREMENT
                                        ERROR   \N
004762  012767  000065  174074  ERR65:  MOV     #65,ERCOUNT     ;SET UP ERROR COUNT 65
        000066                          N=N+1
004770  004567  003502                  JSR     %5,STAER1       ;REPORT CMA INCREMENTED
004774  000717                          BR      INHCMA          ;LOOP ON ERROR
004776  000004                  LPINH:  SCOPE
005000  004634                          INHCMA

                                        ;
                                        ;IN THIS STATIC TEST THE ABILITY OF THE DISK CONTROL
                                        ;TO INCREMENT THE SURFACE REGIST WILL BE TESTED.
                                        ;
                                        ;A ONE WORD WRITE WILL BE EXECUTED
                                        ;RKDA=13
                                        ;WC=-1
                                        ;CMA=OUTBUF
                                        ;
                                        ;AT THE COMPLETION OF THE WRITE
                                        ;RKDA=20
                                        ;
005002  012777  000001  174004  STAI74: MOV     #BIT0,@DCS      ;CLEAR THE DISK
005010  105777  174000                  TSTB    @DCS
005014  100375                          BPL     .-4
005016  012777  177777  173772          MOV     #177777,@WC     ;SET WORD COUNT TO -1
005024  012777  011144  173766          MOV     #OUTBUF,@CMA    ;SET UP CURRENT ADDRESS
005032  012777  000013  173762          MOV     #13,@DAR        ;SET RKDA TO 13(LAST SECTOR ON SURFACE 0)
```

```
005040  012777  000003  173746          MOV     #3,@DCS         ;GO WRITE ONE WORD
005046  005067  174022                  CLR     WRD             ;LET'S WAIT FOR COMPLETION
005052  105777  173736          LTSWT:  TSTB    @DCS            ;TEST FOR NOT BUSY
005056  100414                  BMI     STAI75          ;X-FER COMPLETE
005060  005267  174010                  INC     WRD             ;+1 WAIT REG,
005064  001372                  BNE     LTSWT           ;GO WAIT FOR NOT BUSY
005066  017767  173722  174000          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                        ERROR   \N
005074  012767  000066  173762  ERR66:  MOV     #66,ERCOUNT     ;SET UP ERROR COUNT 66
        000067                          N=N+1
005102  004567  003316                  JSR     %5,STAER        ;REPORT ERROR X-FER NEVER COMPLETED
005106  000735                  BR      STAI74          ;RESTART ROUTINE
005110  022777  000020  173704  STAI75: CMP     #20,@DAR        ;DID THE SURFACE INC
005116  001412                  BEQ     LPST74          ;RKDA OK NO ERROR
005120  017767  173676  173746          MOV     @DAR,WRD        ;FETCH DAR
                                        ERROR   \N
005126  012767  000067  173730  ERR67:  MOV     #67,ERCOUNT     ;SET UP ERROR COUNT 67
        000070                          N=N+1
005134  004567  003264                  JSR     %5,STAER        ;REPORT DAR DID NOT INC TO 0
005140  000167  177636                  JMP     STAI74          ;ERROR RE-START ROUTINE
005144  000004          LPST74: SCOPE                           ;ENTER SCOPE LOOP
005146  005002                  STAI74
```

```
                                ;
                                ;IN THIS ROUTINE THE ABILITY OF THE CONTROL
                                ;TO INCREMENT THE CYLINDER IS CHECKED
                                ;
                                ;ROUTINE PARAMETERS AT START
                                ;RKDA=33
                                ;WC=-1
                                ;CMA=#OUTBUF
                                ;DCS=#3
                                ;
                                ;PARAMETERS AT THE END OF ROUTINE
                                ;RKDA=40
                                ;WC=0
                                ;CMA=OUTBUF+2
                                ;DCS=0
                                ;
                                ;
005150  012777  000033  173644  STAI77: MOV     #33,@DAR        ;RKDA EQUALS THE LAST CYLINDER LAST SECTOR
005156  012777  177777  173632          MOV     #177777,@WC     ;WORD COUNT SET FOR ONE WORD
005164  012777  011144  173626          MOV     #OUTBUF,@CMA    ;CURRENT ADDRESS SET UP
005172  052777  000003  173614          BIS     #3,@DCS         ;GO WRITE
005200  105777  173610                  TSTB    @DCS            ;IS RDY CLEARED
005204  100011                          BPL     STA100          ;
005206  017767  173602  173660          MOV     @DCS,WRD        ;FETCH CONTROL RUG
                                        ERROR   \N
005214  012767  000070  173642  ERR70:  MOV     #70,ERCOUNT     ;SET UP ERROR COUNT 70
        000071                          N=N+1
005222  004567  003176                  JSR     %5,STAER        ;REPORT BUSY NOT SET
005226  000750                          BR      STAI77          ;RESTART ROUTINE
005230  005067  173640          STA100: CLR     WRD
005234  105777  173554          INXDSK: TSTB    @DCS            ;IS X-FER COMPLETE
005240  100415                          BMI     STA101
005242  005267  173626                  INC     WRD             ;TEST READY SHOULD BE SET
005246  001372                          BNE     INXDSK          ;GO CHECK AGAIN
005250  017767  173540  173616          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                        ERROR   \N
005256  012767  000071  173600  ERR71:  MOV     #71,ERCOUNT     ;SET UP ERROR COUNT 71
        000072                          N=N+1
005264  004567  003134                  JSR     %5,STAER        ;REPORT X-FER NOT COMPLETE
005270  000167  177654                  JMP     STAI77          ;RESTART ROUTINE
005274  022777  000040  173520  STA101: CMP     #40,@DAR        ;DID THE DRIVE NUMBER INC
005302  001412                          BEQ     LPST77          ;YES
005304  017767  173512  173562          MOV     @DAR,WRD        ;FETCH DAR ITS NOT CLEAR
                                        ERROR   \N
005312  012767  000072  173544  ERR72:  MOV     #72,ERCOUNT     ;SET UP ERROR COUNT 72
        000073                          N=N+1
005320  004567  003100                  JSR     %5,STAER        ;REPORT DAR NOT CLEAR.
005324  000167  177620                  JMP     STAI77          ;RESTART ROUTINE
005330  000004          LPST77: SCOPE                           ;ENTER SCOPE LOOP
005332  005150                  STAI77
```

```
                                ;
                                ;
                                ;IN THIS ROUTINE THE PROGRAM WILL GENERATE A
                                ;OVERRUN ERROR THIS WILL BE DONE
                                ;BY ATTEMPING A 2 BLOCK WRITE ON
                                ;THE LAST CYLINDER AND THE LAST SECTOR ON THE
                                ;
                                ;
005334  012777  000001  173452  STA103: MOV     #BIT0,@DCS      ;CLEAR THE DISK
005342  105777  173446          TSTB    @DCS
005346  100375                  BPL     .-4
005350  012777  014533  173444          MOV     #14533,@DAR     ;SET DAR=TO ALL ONES
005356  012777  177000  173432          MOV     #-1000,@WC      ;WORD COUNT=(2 BLOCKS ON HI DENSITY DRIVE)
005364  012777  011144  173426          MOV     #OUTBUF,@CMA    ;CURRENT ADDRESS=OUTBUF
005372  012777  000003  173414          MOV     #3,@DCS         ;GO WRITE
005400  005067  173470                  CLR     WRD
005404  105777  173404          WAFBUS: TSTB    @DCS            ;IS RDY SET
005410  100414                  BMI     STX104          ;YES EXIT
005412  005267  173456          INC     WRD             ;NO BUSY STILL SET
005416  001372                  BNE     WAFBUS          ;GO WAIT FOR BUSY=0
005420  017767  173370  173446          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                ERROR   \N
005426  012767  000073  173430  ERR73:  MOV     #73,ERCOUNT     ;SET UP ERROR COUNT 73
        000074                  N=N+1
005434  004567  002764                  JSR     %5,STAER        ;REPORT BUSY SET
005440  000735                  BR      STA103          ;RESTART ROUTINE
005442  032777  040000  173354  STX104: BIT     #BIT14,@RKER    ;DID THE CONTROL OVERRUN .
005450  001014                  BNE     STA104          ;YES BRANCH
005452  017767  173336  173416          MOV     @DCS,WRD1
005460  017767  173340  173406          MOV     @RKER,WRD
                                ERROR   \N
005466  012767  000074  173370  ERR74:  MOV     #74,ERCOUNT     ;SET UP ERROR COUNT 74
        000075                  N=N+1
005474  004567  002776                  JSR     %5,STAER1       ;REPORT DRIVE #DID NOT INC.
005500  000715                  BR      STA103          ;LOOP ON ERROR
005502  000004          .       STA104: SCOPE
005504  005334                  STA103
                                ;
                                ;
                                ;
                                ;TEST FOR NON-EXISTENT DRIVE ERROR
                                ;THIS TEST WILL REPORT ERRORS ON A MAX. DRIVE SYSTEM
                                ;8 DRIVES IS A MAX. DRIVE SYSTEM
005506  012777  000001  173300  NXDST:  MOV     #BIT0,@DCS      ;CLEAR THE DRIVE
005514  105777  173274          TSTB    @DCS
005520  100375                  BPL     .-4
005522  012777  177777  173266          MOV     #-1,@WC         ;SET UP THE WORD COUNT
005530  012777  160000  173264          MOV     #160000,@DAR    ;LOAD ADDRESS REG TO DRIVE ONE
005536  012777  000003  173250          MOV     #3,@DCS         ;GO WRITE
005544  105777  173244          TSTB    @DCS
005550  100375                  BPL     .-4
005552  005777  173236          TST     @DCS            ;IS THE ERROR BIT SET
005556  100413                  BMI     ERNXD           ;YES THE ERROR BIT IS SET
```

```
005560  017767  173230  173310          MOV     @DCS,WRD1
005566  017767  173232  173300          MOV     @RKER,WRD
                                        ERROR   \N
005574  012767  000075  173262  ERR75:  MOV     #75,ERCOUNT     ;SET UP ERROR COUNT 75
        000076                          N=N+1
005602  004567  002670                  JSR     %5,STAER1       ;REPORT ERROR BIT NOT SET
005606  032777  000200  173210  ERNXD:  BIT     #BIT7,@RKER     ;IS NCD SET
005614  001010                          BNE     STA105          ;YES; IS ERROR SET
005616  017767  173172  173250          MOV     @DCS,WRD        ;FETCH DCS
                                        ERROR   \N
005624  012767  000076  173232  ERR76:  MOV     #76,ERCOUNT     ;SET UP ERROR COUNT 76
        000077                          N=N+1
005632  004567  002566                  JSR     %5,STAER        ;REPORT NEED NOT SET
005636  005777  173152          STA105: TST     @DCS            ;IS ERROR FLAG SET
005642  100410                          BMI     LPSX103         ;ERROR IS SET
005644  017767  173144  173222          MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
                                        ERROR   \N
005652  012767  000077  173204  ERR77:  MOV     #77,ERCOUNT     ;SET UP ERROR COUNT 77
        000100                          N=N+1
005660  004567  002540                  JSR     %5,STAER        ;REPORT ERROR NOT SET
005664  000004          LPSX103:SCOPE                           ;ENTER SCOPE LOOP
```

```
005666  005506                          NXDST:
                                        ;
                                        ;TEST NON-EXISTENT SECTOR ERROR
                                        ;
005670  012777  000001  173116  ST103X: MOV     #BIT0,@DCS      ;CLEAR THE CONTROL
005676  105777  173112                  TSTB    @DCS
005702  100375                          BPL     .-4
005704  012777  000017  173110          MOV     #17,@DAR        ;LOAD A NON-EXISTENT SECTOR
005712  012777  177777  173076          MOV     #-1,@WC         ;LOAD THE WORD COUNT
005720  012777  000003  173066          MOV     #3,@DCS         ;GO WRITE
005726  105777  173062                  TSTB    @DCS
005732  100375                          BPL     .-4
005734  005777  173054                  TST     @DCS            ;TEST FOR ERROR
005740  100414                          BMI     LP103X          ;BRANCH IF ERROR SET
005742  017767  173054  173124          MOV     @DAR,WRD
005750  017767  173040  173120          MOV     @DCS,WRD1
                                        ERROR   \N
005756  012767  000100  173100  ERR100: MOV     #100,ERCOUNT    ;SET UP ERROR COUNT 100
        000101                          N=N+1
005764  004567  002506                  JSR     %5,STAER1       ;REPORT ERROR NOT SET
005770  000737                          BR      ST103X          ;LOOP ON ERROR
005772  032777  000040  173024  LP103X: BIT     #BIT5,@RKER     ;TEST FOR NXS SET
006000  001014                          BNE     X103LP          ;BRANCH NXS SET
006002  017767  173016  173064          MOV     @RKER,WRD
006010  017767  173000  173060          MOV     @DCS,WRD1
                                        ERROR   \N
006016  012767  000101  173040  ERR101: MOV     #101,ERCOUNT    ;SET UP ERROR COUNT 101
        000102                          N=N+1
006024  004567  002446                  JSR     %5,STAER1       ;REPORT NXS NOT SET
006030  000717                          BR      ST103X          ;LOOP ON ERROR
006032  000004                  X103LP: SCOPE
006034  005670                          ST103X
                                        ;
                                        ;
                                        ;TEST NON-EXISTENT CYLINDER
                                        ;
006036  012777  000001  172750  ST104X: MOV     #BIT0,@DCS      ;CLEAR THE CONTROL
006044  105777  172744                  TSTB    @DCS
006050  100375                          BPL     .-4
006052  012777  017740  172742          MOV     #17740,@DAR     ;LOAD NON-EXISTENT CYLINDER
006060  012777  177777  172730          MOV     #-1,@WC         ;LOAD WORD COUNT
006066  012777  000003  172720          MOV     #3,@DCS         ;GO WRITE
006074  105777  172714                  TSTB    @DCS
006100  100375                          BPL     .-4
006102  005777  172706                  TST     @DCS            ;TEST FOR ERROR BIT SET
006106  100414                          BMI     LP104X          ;BRANCH ERROR BIT SET
006110  017767  172706  172756          MOV     @DAR,WRD
006116  017767  172672  172752          MOV     @DCS,WRD1
                                        ERROR   \N
006124  012767  000102  172732  ERR102: MOV     #102,ERCOUNT    ;SET UP ERROR COUNT 102
        000103                          N=N+1
006132  004567  002340                  JSR     %5,STAER1       ;REPORT ERROR BIT NOT SET
006136  000737                          BR      ST104X          ;LOOP ON ERROR
006140  032777  000100  172656  LP104X: BIT     #BIT6,@RKER     ;TEST FOR NXC SET
006146  001014                          BNE     X104LP          ;ERROR BIT SET OK
```

```
006150  017767  172650  172716          MOV     @RKER,WRD
006156  017767  172632  172712          MOV     @DCS,WRD1
                                        ERROR   \N
006164  012767  000103  172672  ERR103: MOV     #103,ERCOUNT    ;SET UP ERROR COUNT 103
        000104                          N=N+1
006172  004567  002300                  JSR     %5,STAER1       ;REPORT NXC NOT SET
006176  000717                          BR      ST104X          ;LOOP ON ERROR
006200  000004                  X104LP: SCOPE
006202  006036                          ST104X
                                ;
                                ;
                                ;ROUTINE TO TEST PROGRAM ERROR
                                ;
                                ;
006204  012777  000001  172602  ST105X: MOV     #BIT0,@DCS      ;CLEAR THE CONTROL
006212  105777  172576                  TSTB    @DCS
006216  100375                          BPL     .-4
006220  012777  002015  172566          MOV     #2015,@DCS      ;ISSUE SEEK WITH BIT10 SET
006226  105777  172562                  TSTB    @DCS            ;TEST FOR CONTROL READY
006232  100375                          BPL     .-4             ;WAIT FOR CONTROL READY
006234  005777  172554          LPX105: TST     @DCS            ;IS THE ERROR BIT SET
006240  100411                          BMI     ST106X          ;ERROR SET OK BRANCH
006242  017767  172546  172624          MOV     @DCS,WRD        ;REPORT DCS
                                        ERROR   \N
006250  012767  000104  172606  ERR104: MOV     #104,ERCOUNT    ;SET UP ERROR COUNT 104
        000105                          N=N+1
006256  004567  002142                  JSR     %5,STAER        ;REPORT ERROR NOT SET
006262  000750                          BR      ST105X
006264  032777  004000  172532  ST106X: BIT     #BIT11,@RKER    ;TEST FOR PROGRAMMING ERROR
006272  001015                          BNE     LPX106          ;BIT SET OK
006274  017767  172514  172574          MOV     @DCS,WRD1
006302  017767  172516  172564          MOV     @RKER,WRD
                                        ERROR   \N
006310  012767  000105  172546  ERR105: MOV     #105,ERCOUNT    ;SET UP ERROR COUNT 105
        000106                          N=N+1
006316  004567  002154                  JSR     %5,STAER1
006322  000167  177656                  JMP     ST105X          ;LOOP ON ERROR
006326  000004                  LPX106: SCOPE
006330  006204                          ST105X
```

```
                                ;
                                ;IN THIS ROUTINE THE ABILITY OF NON-EXISTENT
                                ;MEMORY ERROR IS CHECK.
                                ;
                                ;TRANSFER TWO WORDS STARTING WITH THE
                                ;LARGEST ADDRESSABLE MEMORY LOCATION IN THE
                                ;PDP-11
                                ;
                                ;
006332  012777  000001  172454  NXMTSM: MOV     #BIT0,@DCS      ;CLEAR THE DISK
006340  105777  172450          TSTB    @DCS
006344  100375                  BPL     .-4
006346  012777  000340  172426          MOV     #340,@CSR       ;LOCK UP PROCESSOR
006354  012777  177776  172434          MOV     #-2,@WC         ;SET UP WORD COUNT
006362  012777  177776  172430          MOV     #177776,@CMA    ;SET UP CURRENT ADDRESS
006370  012777  000063  172416          MOV     #63,@DCS        ;ISSUE WRITE
006376  105777  172412                  TSTB    @DCS            ;TEST FOR READY
006402  100375                  BPL     .-4
006404  005777  172404                  TST     @DCS            ;TEST FOR ERROR FLAG
006410  100414                  BMI     TSTNEM          ;ERROR BIT SET OK
                                ERROR   \N
006412  012767  000106  172444  ERR106: MOV     #106,ERCOUNT    ;SET UP ERROR COUNT 106
        000107                  N=N+1
006420  017767  172370  172450          MOV     @DCS,WRD1       ;FETCH FUNCTION REG
006426  017767  172372  172440          MOV     @RKER,WRD       ;FETCH ERROR REG
006434  004567  002036                  JSR     %5,STAER1       ;REPORT ERROR BIT NOT SET
006440  000734                  BR      NXMTSM          ;LOOP ON ERROR
006442  032777  040000  172344  TSTNEM: BIT     #BIT14,@DCS
006450  001011                  BNE     HWDOK           ;BRANCH IF HARD ERROR SET
                                ERROR   \N
006452  012767  000107  172404  ERR107: MOV     #107,ERCOUNT    ;SET UP ERROR COUNT 107
        000110                  N=N+1
006460  017767  172330  172406          MOV     @DCS,WRD        ;REPORT HWD NOT SET BY NEM
006466  004567  001732                  JSR     %5,STAER        ;REPORT HWD NOT SET
006472  000717                  BR      NXMTSM          ;LOOP ON ERROR
006474  032777  002000  172322  HWDOK:  BIT     #BIT10,@RKER    ;TEST FOR NEM SET
006502  001011                  BNE     CLRNEM          ;NEM SET BE X-FER OK
                                ERROR   \N
006504  012767  000110  172352  ERR110: MOV     #110,ERCOUNT    ;SET UP ERROR COUNT 110
        000111                  N=N+1
006512  017767  172306  172354          MOV     @RKER,WRD       ;REPORT HARD ERROR REGISTER
006520  004567  001700                  JSR     %5,STAER        ;REPORT NEM NOT SET BY X-FER
006524  000702                  BR      NXMTSM          ;LOOP ON ERROR
006526  012777  000001  172260  CLRNEM: MOV     #BIT0,@DCS      ;CLEAR THE DISK
006534  105777  172254          TSTB    @DCS            ;IS THE ERROR CLEARED
006540  100375                  BPL     .-4             ;WAIT FOR READY
006542  032777  040000  172244  CKHWD:  BIT     #BIT14,@DCS     ;IS HARD ERROR SET
006550  001412                  BEQ     CKNEM           ;HWD CLEARED BY DISK CLEAR
                                ERROR   \N
006552  012767  000111  172304  ERR111: MOV     #111,ERCOUNT    ;SET UP ERROR COUNT 111
        000112                  N=N+1
006560  017767  172230  172306          MOV     @DCS,WRD        ;REPORT HWD NOT CLEARED
006566  004567  001632                  JSR     %5,STAER        ;
006572  000167  177534                  JMP     NXMTSM          ;LOOP ON ERROR
006576  032777  002000  172220  CKNEM:  BIT     #BIT10,@RKER    ;IS NEM CLEARED BY DISK CLEAR
```

```
006604  001410                          BEQ     LPNEM           ;NEM CLEARED BY DISK CLEAR
                                        ERROR   \N
006606  012767  000112  172250  ERR112: MOV     #112,ERCOUNT    ;SET UP ERROR COUNT 112
        000113                          N=N+1
006614  017767  172204  172252          MOV     @RKER,WRD       ;REPORT NEM NOT CLEARED BY DISK CLEAR
006622  000167  177504                  JMP     NXMTSM          ;LOOP ON ERROR
006626  000004                  LPNEM:  SCOPE                   ;ENTER SCOPE LOOP
006630  006332                          NXMTSM
```

```
                                ;
                                ;IN THIS ROUTINE THE PROGRAM WILL TEST
                                ;THAT THE DISK WILL ONLY TRAP
                                ;AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
                                ;BY CLEARING THE DONE
                                ;PROCESSOR OPERATING AT PRIORITY7
                                ;
006632  012777  006732  172174  STA106: MOV     #INT106,@VECTOR ;SET UP INTERRUPT VECTOR
006640  012777  000340  172134          MOV     #340,@CSR       ;LOCK OUT ALL INTERRUPTS
006646  005077  172150                  CLR     @DAR
006652  012777  177777  172136          MOV     #177777,@WC     ;SET WORD COUNT TO -1
006660  012777  011144  172132          MOV     #OUTBUF,@CMA    ;LOAD CURRENT ADDRESS
006666  052777  000100  172120          BIS     #100,@DCS       ;GO  (INTERRUPT ENABLED)
006674  005000                          CLR     %0              ;WAIT FOR RDY
006676  005200                          INC     %0
006700  001376                          BNE     .-2
006702  105777  172106                  TSTB    @DCS            ;IS CONTROL STILL NOT RDY
006706  100423                          BMI     LP106           ;NO
006710  017767  172100  172156          MOV     @DCS,WRD        ;YES!
                                        ERROR   \N
006716  012767  000113  172140  ERR113: MOV     #113,ERCOUNT    ;SET UP ERROR COUNT 113
        000114                          N=N+1
006724  004567  001474                  JSR     %5,STAER        ;REPORT CONTROL STILL BUSY
006730  000740                          BR      STA106          ;RESTART ROUTINE
                                ;
                                ;PROCESSOR SHOULD NOT TRAP TO INT106
                                ;PROCESSOR PRIORITY IS LOCKED AT 7
                                ;
                                INT106: ERROR   \N
006732  012767  000114  172124  ERR114: MOV     #114,ERCOUNT    ;SET UP ERROR COUNT 114
        000115                          N=N+1
006740  017767  172036  172126          MOV     @CSR,WRD        ;FETCH PROCESSOR PRIORITY
006746  022626                          CMP     (6)+,(6)+
006750  004567  001450                  JSR     %5,STAER        ;REPORT DISK INTERRUPTED
006754  000726                          BR      STA106
006756  000004                  LP106:  SCOPE                   ;ENTER SCOPE LOOP
006760  006632                          STA106
```

```
                                ;
                                ;IN THIS ROUTINE THE PROGRAM WILL TEST
                                ;THAT THE DISK WILL ONLY TRAP
                                ;AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
                                ;BY CLEARING THE DONE
                                ;PROCESSOR OPERATING AT PRIORITY6
                                ;
006762  012777  007062  172044 STA110: MOV     #INT110,@VECTOR ;SET UP INTERRUPT VECTOR
006770  012777  000300  172004         MOV     #300,@CSR       ;LOCK OUT ALL INTERRUPTS ABOVE
006776  005077  172020                 CLR     @DAR
007002  012777  177777  172006         MOV     #177777,@WC     ;SET WORD COUNT TO -1
007010  012777  011144  172002         MOV     #OUTBUF,@CMA    ;LOAD CURRENT ADDRESS
007016  012777  000103  171770         MOV     #103,@DCS       ;GO WRITE (INTERRUPT ENABLED)
007024  005000                 CLR     %0              ;WAIT FOR NOT BUSY
007026  005200                 INC     %0
007030  001376                 BNE     .-2
007032  105777  171756                 TSTB    @DCS            ;IS CONTROL STILL BUSY
007036  100423                 BMI     LP110           ;NO
007040  017767  171750  172026         MOV     @DCS,WRD        ;YES!
                                ERROR   \N
007046  012767  000115  172010 ERR115: MOV     #115,ERCOUNT    ;SET UP ERROR COUNT 115
        000116                  N=N+1
007054  004567  001344                 JSR     %5,STAER        ;REPORT CONTROL STILL BUSY
007060  000740                 BR      STA110          ;RESTART ROUTINE
                                ;
                                ;PROCESSOR SHOULD NOT TRAP TO INT106
                                ;PROCESSOR PRIORITY IS LOCKED AT 6
                                ;
                                INT110: ERROR   \N
007062  012767  000116  171774 ERR116: MOV     #116,ERCOUNT    ;SET UP ERROR COUNT 116
        000117                  N=N+1
007070  017767  171706  171776         MOV     @CSR,WRD        ;FETCH PROCESSOR PRIORITY
007076  022626                 CMP     (6)+,(6)+
007100  004567  001320                 JSR     %5,STAER        ;REPORT DISK INTERRUPTED
007104  000726                 BR      STA110
                                ;
007106  000004                 LP110:  SCOPE                   ;ENTER SCOPE LOOP
007110  006762                 STA110
                                ;
                                ;IN THIS ROUTINE THE PROGRAM WILL TEST
                                ;THAT THE DISK WILL ONLY TRAP
                                ;AT BR5 ONLY WHEN A INTERRUPT IS GENERATED
                                ;BY CLEARING THE DONE
                                ;PROCESSOR OPERATING AT PRIORITY5
                                ;
007112  012777  007212  171714 STA112: MOV     #INT112,@VECTOR ;SET UP INTERRUPT VECTOR
007120  012777  000240  171654         MOV     #240,@CSR       ;LOCK OUT ALL INTERRUPTS ABOVE
007126  005077  171670                 CLR     @DAR
007132  012777  177777  171656         MOV     #177777,@WC     ;SET WORD COUNT TO -1
007140  012777  011144  171652         MOV     #OUTBUF,@CMA    ;LOAD CURRENT ADDRESS
007146  012777  000105  171640         MOV     #105,@DCS       ;GO READ (INTERRUPT ENABLED)
007154  005000                 CLR     %0              ;WAIT FOR NOT BUSY
007156  005200                 INC     %0
007160  001376                 BNE     .-2
```

```
007162  105777  171626          TSTB    @DCS            ;IS CONTROL STILL BUSY
007166  100423                  BMI     LP112           ;NO
007170  017767  171620  171676 MOV     @DCS,WRD        ;YES!
                                ERROR   \N
007176  012767  000117  171660 ERR117: MOV     #117,ERCOUNT    ;SET UP ERROR COUNT 117
        000120                  N=N+1
007204  004567  001214          JSR     %5,STAER        ;REPORT CONTROL STILL BUSY
007210  000740                  BR      STA112          ;RESTART ROUTINE
```

```
                                ;
                                ;PROCESSOR SHOULD NOT TRAP TO INT106
                                ;PROCESSOR PRIORITY IS LOCKED AT 7
                                ;
                                INT112: ERROR   \N
007212  012767  000120  171644 ERR120: MOV     #120,ERCOUNT    ;SET UP ERROR COUNT 120
        000121                  N=N+1
007220  017767  171556  171646         MOV     @CSR,WRD        ;FETCH PROCESSOR PRIORITY
007226  022626                 CMP     (6)+,(6)+
007230  004567  001170         JSR     %5,STAER        ;REPORT DISK INTERRUPTED
007234  000726                 BR      STA112
                                ;
007236  000004                 LP112:  SCOPE                   ;ENTER SCOPE LOOP
007240  007112                 STA112
                                ;
                                ;
                                ;
                                ;THIS ROUTINE WILL TEST THE ABILITY OF THE DISK CONTROL
                                ;TO TRAP AT BR5 WHEN THE DONE FLAG IS SET.
                                ;
007242  012777  000001  171544 STA114: MOV     #BIT0,@DCS      ;CLEAR THE DISK
007250  105777  171540         TSTB    @DCS
007254  100375                 BPL     .-4
007256  012777  007350  171550         MOV     #INT114,@VECTOR ;SET UP DISK TRAP VECTOR
007264  012777  000200  171510         MOV     #200,@CSR       ;SET PROCESSOR TO PRIORITY 4
007272  005077  171524         CLR     @DAR            ;CLEAR LOW ORDER ADDRESS BITS
007276  012777  177777  171512         MOV     #177777,@WC     ;SET WORD COUNT TO -1
007304  012777  011144  171506         MOV     #OUTBUF,@CMA    ;LOAD CURRENT ADDRESS
007312  012777  000107  171474         MOV     #107,@DCS       ;WRITE CHECK (INTERRUPT ENABLE
007320  005000                 CLR     %0
007322  005200                 INC     %0              ;WAIT FOR INTERRUPT TO OCCUR
007324  001376                 BNE     .-2
                                ERROR   \N
007326  012767  000121  171530 ERR121: MOV     #121,ERCOUNT    ;SET UP ERROR COUNT 121
        000122                  N=N+1
007334  017767  171454  171532         MOV     @DCS,WRD        ;FETCH CONTENTS OF DCS
007342  004567  001056         JSR     %5,STAER        ;REPORT INTERRUPT NO INTERRUPT
007346  000735                 BR      STA114          ;RESTART ROUTINE
                                ;
                                ;
                                ;
007350  022626                 INT114: CMP     (6)+,(6)+       ;TRAP OK
007352  000004                 SCOPE                   ;ENTER SCOPE LOOP
007354  007242                 STA114
                                ;
                                ;
                                ;TEST RESTORE DRIVE FUNCTION
                                ;
007356  012777  000001  171430 RESTORE:MOV     #BIT0,@DCS      ;CLEAR THE CONTROL
007364  105777  171424         TSTB    @DCS
007370  100375                 BPL     .-4
007372  005077  171404         CLR     @CSR            ;CLEAR CPU PRIORITY
007376  012777  007422  171430         MOV     #RDYRES,@VECTOR ;SET UP INTERRUPT VECTOR
```

```
007404  012777  000340  171424          MOV     #340,@STATUS
007412  012777  000115  171374          MOV     #115,@DCS       ;ISSUE RESTORE
007420  000001                          WAIT
007422  105777  171366          RDYRES: TSTB    @DCS            ;IS THE DRIVE READY
007426  100411                          BMI     TSTRES          ;BRANCH IF READY IS SET
                                        ERROR   \N
007430  012767  000122  171426  ERR122: MOV     #122,ERCOUNT    ;SET UP ERROR COUNT 122
        000123                          N=N+1
007436  017767  171352  171430          MOV     @DCS,WRD        ;FETCH FUNCTION REG
007444  004567  000754                  JSR     %5,STAER        ;REPORT READY NOT SET
007450  000742                          BR      RESTORE         ;LOOP ON ERROR
007452  032777  000100  171346  TSTRES: BIT     #BIT6,@RKDS     ;TEST ARE THE HEADS MOVING
007460  001414                          BEQ     RDYINT          ;BRANCH IF THE HEADS ARE MOVING
007462  017767  171326  171406          MOV     @DCS,WRD1
007470  017767  171332  171400          MOV     @RKDS,WRD1
                                        ERROR   \N
007476  012767  000123  171360  ERR123: MOV     #123,ERCOUNT    ;SET UP ERROR COUNT 123
        000124                          N=N+1
007504  004567  000766                  JSR     %5,STAER1       ;REPORT HEADS NOT MOVING
007510  000722                          BR      RESTORE         ;LOOP ON ERROR
007512  022626                  RDYINT: CMP     (6)+,(6)+       ;RESET STACK
007514  012777  007530  171312          MOV     #THREST,@VECTOR ;SETUP VECTOR ADDRESS
007522  005077  171254                  CLR     @CSR            ;CLEAR STATUS
007526  000001                          WAIT
007530  022626                  THREST: CMP     (6)+,(6)+       ;RESET THE STACK
007532  032777  020000  171254          BIT     #BIT13,@DCS     ;TEST FOR SEARCH COMPLETE
007540  001773                          BEQ     THREST          ;WAIT FOR DRIVE READY
007542  005777  171246                  TST     @DCS            ;IS THERE AN ERROR
007546  100014                          BPL     LPREST          ;NO ERROR RESTORE OK
007550  017767  171240  171320          MOV     @DCS,WRD1
007556  017767  171244  171310          MOV     @RKDS,WRD
                                        ERROR   \N
007564  012767  000124  171272  ERR124: MOV     #124,ERCOUNT    ;SET UP ERROR COUNT 124
        000125                          N=N+1
007572  004567  000700                  JSR     %5,STAER1       ;REPORT ERROR WHILE RESTORING
007576  000667                          BR      RESTORE         ;LOOP ON ERROR
007600  000004                  LPREST: SCOPE
007602  007356                          RESTORE
                                ;* * * ADDRESS TEST 1 * * *
                                ;
                                ;
                                ;READ SECTOR FORMAT AND VERIFTY THE CYLINDER
                                ;ADDRESS IS CORRECT
                                ;
                                ;NOTE: DATA IS NOT CHECK IN TEST
                                ;
                                ;
007604  012706  001000          ADT1:   MOV     #1000,%6        ;SET UP THE STACK
007610  005067  171234                  CLR     DMA
007614  016777  171230  171200  RDFMT:  MOV     DMA,@DAR        ;LOAD THE DRIVE ADDRESS
007622  012777  177752  171166          MOV     #-26,@WC                ;LOAD WORD COUNT
007630  012777  011144  171162          MOV     #OUTBUF,@CMA    ;LOAD CURRENT ADDRESS
007636  012777  002005  171150          MOV     #2005,@DCS      ;LOAD READ FORMAT
007644  105777  171144                  TSTB    @DCS            ;TEST FOR CONTROL READY
007650  100375                          BPL     .-4
```

```
007652  005777  171136                   TST     @DCS            ;TEST FOR ERROR
007656  100014                           BPL     NOFMT           ;BRANCH IF NO ERROR
007660  017767  171130  171210           MOV     @DCS,WRD1
007666  017767  171130  171200           MOV     @DAR,WRD
                                         ERROR   \N
007674  012767  000125  171162  ERR125: MOV     #125,ERCOUNT    ;SET UP ERROR COUNT 125
        000126                           N=N+1
007702  004567  000570                   JSR     %5,STAER1       ;REPORT ERROR OCCURRED
007706  000742                           BR      RDFMT           ;LOOP ON ERROR
007710  005067  171152           NOFMT:  CLR     SAVE
007714  016767  171130  171154           MOV     DMA,WRD1
007722  012701  011144                   MOV     #OUTBUF,%1
007726  026711  171144           CMPNEX: CMP     WRD1,@%1        ;IS THE FORMAT CORRECT
007732  001411                           BEQ     CHKNEX          ;FORMAT CORRECT BRANCH
007734  016767  171126  171132           MOV     SAVE,WRD
                                         ERROR   \N
007742  012767  000126  171114  ERR126: MOV     #126,ERCOUNT    ;SET UP ERROR COUNT 126
        000127                           N=N+1
007750  004567  000522                   JSR     %5,STAER1       ;REPORT SECTOR IN ERRROR
007754  000717                           BR      RDFMT           ;CHECK NEXT SECTOR
007756  005267  171104           CHKNEX: INC     SAVE
007762  022767  000027  171076           CMP     #27,SAVE        ;HAVE ALL SECTORS BEEN CHECKED
007770  001356                           BNE     CMPNEX          ;NO CHECK NEXT ONE
007772  022767  014500  171050           CMP     #14500,DMA      ;IS THIS THE LAST CYLINDER
010000  001404                           BEQ     LPADT           ;YES EXIT
010002  062767  000040  171040           ADD     #40,DMA         ;SET UP NEXT CYLINDER
010010  000701                           BR      RDFMT
010012  032777  004000  170760  LPADT:  BIT     #BIT11,@SR      ;LOOP ON TEST
010020  001402                           BEQ     .+6
010022  000167  177556                   JMP     ADT1            ;LOOP ON THE TEST
                                ;
                                ;
                                ;
                                ;
                                ;

010026  104001                           EMT+1                   ;REPORT END
010030  011136                           END
010032  000167  171154                   JMP     STA11           ;RESTART TEST
                                ;
                                ;SCOPE LOOP ROUTINE
                                ;IF BIT 11 SET LOOP ON TEST
                                ;
010036  032777  004000  170734  LOOP:   BIT     #BIT11,@SR      ;TST FOR BIT 11
010044  001402                           BEQ     .+6             ;BIT 11 NOT SET
010046  013646                           MOV     @(6)+,-(6)
010050  000002                           RTI                     ;BIT 11 SET; LOOP ON TEST
010052  005767  171014                   TST     PASS            ;TEST TO SET UP PASS COUNT
010056  001003                           BNE     .+10            ;PASS COUNT SET
010060  012767  000025  171004           MOV     #25,PASS        ;SET UP PASS COUNT
010066  005367  171000                   DEC     PASS            ;SUB. -1 EACH PASS
010072  001402                           BEQ     .+6
010074  013646                           MOV     @(6)+,-(6)
010076  000002                           RTI                     ;LOOP ON TEST
010100  062716  000002                   ADD     #2,(6)          ;INDEX POINTER FOR NEXT TEST
```

```
010104  000002                    RTI                 ;EXIT TO NEXT TEST
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
                                        ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
                                        ;BYTE OF THE CONTROL REGISTER.
                                        ;
010106  016705  170716          DISK:   MOV     DBR,%5          ;SET UP TO LOAD DISK REG
010112  016745  170730                  MOV     TRACK,-(5)      ;LOAD TRACK NUMBER
010116  016745  170726                  MOV     DMA,-(5)        ;LOAD WORD ADDRESS
010122  016745  170726                  MOV     BUF,-(5)        ;SET UP CURRENT ADDRESS
010126  016745  170712                  MOV     WRDCT,-(5)      ;LOAD WORD COUNT
010132  005115                          COM     (5)             ;SET UP TWO'S COMPLEMENT
010134  005215                          INC     (5)
010136  011604                          MOV     (6),%4
010140  014467  170730                  MOV     -(4),WRD        ;
010144  042767  177600  170722          BIC     #177600,WRD     ;MASK FUNCTION BITS
010152  016745  170716                  MOV     WRD,-(5)        ;LOAD FUNCTION REG.
010156  000002                          RTI                     ;RETURN FROM TRAP
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
```

```
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
                                ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
                                ;
                                ;WORD COUNT REGISTER
010160  017777  170614  170630  SELWC:  MOV     @SR,@WC         ;MOV SR INTO WORD COUNT REG
010166  000774                          BR      SELWC
                                ;
                                ;CURRENT ADDRESS REGISTER
010170  017777  170604  170622  SELCMA: MOV     @SR,@CMA        ;MOV SR INTO CURRENT ADDR REG
010176  000774                          BR      SELCMA
                                ;
                                ;DISK ADDRESS REGISTER
010200  017777  170574  170614  SELDAR: MOV     @SR,@DAR        ;MOV SR INTO DISK ADDR REG
010206  000774                          BR      SELDAR
                                ;
                                ;
                                ;DRIVE CONTROL STATUS REGISTER
010210  012777  000340  170564  SELDCS: MOV     #340,@CSR       ;LOCK UP INTERRUPTS
010216  012777  177777  170572          MOV     #177777,@WC     ;SET WORD COUNT =1 WORD
010224  012777  011144  170566          MOV     #OUTBUF,@CMA    ;SET UP CURRENT ADDRESS
010232  017777  170542  170554          MOV     @SR,@DCS        ;MOV SR INTO CONTROL REG
010240  032777  000001  170546          BIT     #BIT0,@DCS      ;IS FUNCTION BITS SET
010246  001760                          BEQ     SELDCS          ;FUNCTION BITS NOT SET
010250  105777  170540          DKBUSY: TSTB    @DCS            ;TEST FOR DRIVE READY
010254  100375                          BPL     DKBUSY          ;DRIVE STILL NOT READY
010256  000754                          BR      SELDCS          ;DRIVE NOT BUSY SELECT NEW CR
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
```

```
                                ;
                                ;THIS ROUTINE ENABLES THE OPERATOR TO SELECT A TRACK STATICLY
                                ;THE ROUTINE DOES A ONE WORD READ TO SELECT THE TRACK
                                ;THE OPERATOR MAY CHANGE THE SWITCH REGISTER AT ANY TIME
                                ;SET DRIVE AND CYLINDER NO. IN SR11-0
                                ;
010260  012777  000001  170526  TKSEL:  MOV     #BIT0,@DCS      ;CLEAR THE CONTROL
010266  105777  170522                  TSTB    @DCS
010272  100375                         BPL     .-4
010274  017767  170500  170574        MOV     @SR,WRD1        ;FETCH SR
010302  016767  170570  170564        MOV     WRD1,WRD
010310  042767  174000  170556        BIC     #174000,WRD     ;MASK DISK AND TRACK NO.
010316  000241                        CLC
010320  006167  170550                ROL     WRD
010324  006167  170544                ROL     WRD
010330  006167  170540                ROL     WRD
010334  006167  170534                ROL     WRD
010340  006167  170530                ROL     WRD
010344  016777  170524  170450        MOV     WRD,@DAR        ;DRIVE ADDRESS REG LOADED
010352  012777  011154  170440        MOV     #INBUF,@CMA     ;LOAD CURRENT ADDRESS
010360  012777  177777  170430        MOV     #177777,@WC     ;LOAD WORD COUNT
010366  052777  000005  170420        BIS     #5,@DCS         ;GO AND READ
010374  032777  000100  170424  CTBUSY: BIT     #BIT6,@RKDS     ;ARE THE HEADS STILL MOVING
010402  100374                        BPL     .-6
010404  105777  170404                TSTB    @DCS            ;TEST FOR CONTROL READY
010410  100371                        BPL     CTBUSY          ;WAIT FOR CONTROL READY
010412  026777  170460  170360  SRCHG:  CMP     WRD1,@SR
010420  001317                        BNE     TKSEL           ;SR HAS CHANGED
010422  000773                        BR      SRCHG           ;SR HAS NOT CHANGED
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
```

```
                                ;
                                ;ROUTINE TO REPORT ERROR COUNT AND CONTENTS OF ONE REGISTER
                                ;
010424  004567  000332          STAER:  JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
010430  001074                          WRD                     ;DATA TO BE CONVERTED
010432  011116                          MES6                    ;ADDRESS OF MESSAGE
010434  000006                          6
010436  004567  000320                  JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
010442  001064                          ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
010444  011052                          HED5                    ;ADDRESS OF MESSAGE
010446  000003                          3
010450  104000                          EMT     +0              ;REPORT MESSAGE
010452  011050                          HED5A
010454  011052                          HED5
010456  011116                          MES6
010460  177777                          -1
010462  032777  002000  170310          BIT     #BIT10,@SR
010470  001401                          BEQ     .+4
010472  000000                          HALT
010474  000205                          RTS     %5              ;EXIT ROUTINE
                                ;
                                ;ROUTINE TO REPORT ERROR COUNT AND THE CONTENTS OF TWO REGISTERS
                                ;
                                ;
                                ;
010476  004567  000260          STAER1: JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
010502  001074                          WRD                     ;DATA TO BE CONVERTED
010504  011116                          MES6                    ;ADDRESS OF MESSAGE
010506  000006                          6
010510  004567  000246                  JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
010514  001076                          WRD1                    ;DATA TO BE CONVERT
010516  011073                          MES5                    ;ADDRESS OF MESSAGE
010520  000006                          6
010522  004567  000234                  JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
010526  001064                          ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
010530  011052                          HED5                    ;ADDRESS OF MESSAGE
010532  000003                          3
010534  104000                          EMT     +0              ;REPORT MESSAGE
010536  011050                          HED5A
010540  011052                          HED5
010542  011073                          MES5
010544  011116                          MES6
010546  177777                          -1
010550  032777  002000  170222          BIT     #BIT10,@SR
010556  001401                          BEQ     .+4
010560  000000                          HALT
010562  000205                          RTS     %5              ;EXIT ROUTINE
                                ;
                                ;
                                ;ROUTINE TO DECODE EMT CALLS
                                ;EMT+1=TYPE ONE LINE OF TEXT
                                ;EMT+0=TYPE A SERIES OF LINES
010564  011600                  EMTRP:  MOV     (6),%0
010566  022740  104001                  CMP     #EMT+1,-(0)     ;WAS THE CALL EMT+1
```

```
010572  001056                  BNE     TYPS            ;NO! TYPE A SERIES OF LINES OF TEXT
010574  000400                  BR      TYP             ;YES TYPE ONE LINE OF TEXT
```

```
                                ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
010576  011600                  TYP:    MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
010600  062716  000002                  ADD     #2,@%6          ;SET UP EXIT.
010604  011000                          MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0.
010606  112067  000114          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
010612  122767  000100  000106          CMPB    #100,TYPDAT     ;CHECK FOR "@" CHARACTER
010620  001001                          BNE     TYPC            ;BRANCH IF NOT "@".
010622  000002                          RTI                     ;TERMINATOR CHAR. DONE. EXIT.
010624  122767  000045  000074  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR "%".
010632  001422                          BEQ     TYPF            ;BRANCH IF "%".
010634  122767  000042  000064          CMPB    #42,TYPDAT      ;NOT "%". CHECK FOR "#".
010642  001423                          BEQ     TYPG            ;BRANCH IF "#"
010644  004767  000002                  JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
010650  000756                          BR      TYPA
010652  032777  040000  170120  TYPD:   BIT     #BIT14,@SR
010660  001006                          BNE     TYEXIT
010662  116777  000040  170114          MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
010670  105777  170114                  TSTB    @TPS            ;WAIT FOR DONE FLAG.
010674  100375                          BPL     .-4
010676  000207                  TYEXIT: RTS     %7              ;EXIT
010700  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
010706  004767  177740                  JSR     %7,TYPD         ;GO TYPE CHAR.
010712  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
010720  004767  177726                  JSR     %7,TYPD         ;GO TYPE CHAR.
010724  000730                          BR      TYPA
010726  000000                  TYPDAT: 0
                                ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
010730  011600                  TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
010732  062716  000002                  ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
010736  011067  000014                  MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
010742  022767  177777  000006          CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
010750  001001                          BNE     TYPSA           ;BRANCH IF NOT TERMINATOR.
010752  000002                          RTI                     ;TERMINATOR, EXIT
010754  104001                  TYPSA:  EMT     +1              ;CALL ON TYP SUB TO TYPE MESSAGE
010756  000000                  TYPSB:  0                       ;ADDRESS OF MESSAGE GOES HERE
010760  000763                          BR      TYPS            ;GO PROCESS NEXT MESSAGE
                                ;
                                ;
                                ;
                                ;
                                ;OCTAL TO ASCII CONVERT ROUTINE
                                ;
                                ;ENTER ROUTINE AS FOLLOWS
                                ;JSR%5,CONV
                                ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED
                                ;ADDR BYTE=LSB OF WHERE ASCII IS GOING
                                ;ASCII#=THE NUMBER OF ASCII CHAR. TO BE CONVERTED
                                ;
                                ;
010762  013567  000054          CONV:   MOV     @(5)+,ACNVX     ;VALUE OF # TO BE CONVERTED
010766  012501                          MOV     (5)+,%1         ;ASCII ADDR
010770  012502                          MOV     (5)+,%2         ;# OF ASCII CHAR
010772  060201                          ADD     %2,%1
```

```
010774  016703  000042          ACVN:   MOV     ACNVX,%3
011000  042703  177770                  BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT OCTAL#
011004  062703  000060                  ADD     #60,%3          ;SET UP ASCII#
011010  110341                          MOVB    %3,-(1)         ;STORE ASCII CHAR
011012  042767  000007  000022          BIC     #7,ACNVX
011020  006067  000016                  ROR     ACNVX           ;ROTATE OCTAL#
011024  006067  000012                  ROR     ACNVX
011030  006067  000006                  ROR     ACNVX
011034  005302                          DEC     %2              ;-1 FROM ASCII CHAR COUNT
011036  001356                          BNE     ACVN
011040  000205                          RTS     %5              ;EXIT # CONVERTED
011042  000000          ACNVX:  0                               ;WRD REGISTER
                        ;
                        ;
                        ;
                        ;
```

```
                                ;
                                ;
                                ;
                                ;
014536  000000                  OUTBUF:0
        000001                          .END
```

```
011112      101
011113      040
011114      040
011115      100

                                ;
                                ;
                                ;
011116      040                 MES6:   .ASCII  /       BAD DATA@/
011117      040
011120      040
011121      040
011122      040
011123      040
011124      040
011125      102
011126      101
011127      104
011130      040
011131      104
011132      101
011133      124
011134      101
011135      100

                                ;
                                ;
                                ;
011136      045                 END:    .ASCII  /%END@/
011137      105
011140      116
011141      104
011142      100
```

```
        011144                          ,EVEN
                                ;
                                ;
                                ;
011144  000000                  OUTBUF: ,WORD   0,0,0,0
011146  000000
011150  000000
011152  000000
                                ;
                                ;
011154  000000                  INBUF:  ,WORD   0,0,0,0
011156  000000
011160  000000
011162  000000
                                ;
                                ;
                                ;
                                ;
        000001                          ,END
```

ACNVX	011042	EROR24	003372	ERR13	001760	ERR71	005256
ACVN	010774	EROR25	003434	ERR14	002030	ERR72	005312
ADT1	007604	EROR26	003502	ERR15	002100	ERR73	005426
BIT0	000001	EROR27	003546	ERR16	002150	ERR74	005466
BIT1	000002	EROR3	002434	ERR17	002224	ERR75	005574
BIT10	002000	EROR30	003606	ERR2	001302	ERR76	005624
BIT11	004000	EROR32	003650	ERR20	002322	ERR77	005652
BIT12	010000	EROR33	003676	ERR21	002376	FLAG	001040
BIT13	020000	EROR34	003740	ERR22	002434	HED5	011052
BIT14	040000	EROR35	004042	ERR23	002476	HED5A	011080
BIT15	100000	EROR36	004116	ERR24	002524	HWDOK	006474
BIT2	000004	EROR37	004154	ERR25	002566	INBUF	011154
BIT3	000010	EROR41	004216	ERR26	002634	INCBUS	002656
BIT4	000020	EROR42	004244	ERR27	002700	INCRD	004376
BIT5	000040	EROR43	004306	ERR3	001332	INCWAT	002342
BIT6	000100	EROR44	004354	ERR30	002740	INHCMA	004634
BIT7	000200	EROR45	004420	ERR31	003002	INT106	006732
BIT8	000400	EROR46	004460	ERR32	003030	INT110	007062
BIT9	001000	EROR5	002476	ERR33	003072	INT112	007212
BUF	001054	EROR50	004522	ERR34	003174	INT114	007350
CHKNEX	007756	EROR51	004550	ERR35	003250	INXDSK	005234
CKHWD	006542	EROR52	004612	ERR36	003302	LOOP	010036
CKNEM	006576	EROR6	002524	ERR37	003344	LP103X	005772
CLRNEM	006526	EROR7	002566	ERR4	001362	LP104X	006140
CMA	001020	ERR0	001232	ERR40	003372	LP106	006756
CMAOK	004736	ERR1	001252	ERR41	003434	LP110	007106
CMPNEX	007726	ERR10	001550	ERR42	003502	LP112	007236
CONV	010762	ERR100	005756	ERR43	003546	LPADT	010012
CSR	001002	ERR101	006016	ERR44	003606	LPINH	004776
CTBUSY	010374	ERR102	006124	ERR45	003650	LPNEM	006626
DAR	001022	ERR103	006164	ERR46	003676	LPREST	007600
DBR	001030	ERR104	006250	ERR47	003740	LPST1	001264
DCS	001014	ERR105	006310	ERR5	001412	LPST10	001562
DISK	010106	ERR106	006412	ERR50	004042	LPST11	001646
DKBUSY	010250	ERR107	006452	ERR51	004116	LPST12	001716
DMA	001050	ERR11	001634	ERR52	004154	LPST13	001772
EMTRP	010564	ERR110	006504	ERR53	004216	LPST14	002042
END	011136	ERR111	006552	ERR54	004244	LPST15	002112
ERCOUN	001064	ERR112	006606	ERR55	004306	LPST16	002162
ERNXD	005606	ERR113	006716	ERR56	004354	LPST17	002236
EROR1	002322	ERR114	006732	ERR57	004420	LPST2	001314
EROR10	002634	ERR115	007046	ERR6	001442	LPST22	003110
EROR11	002700	ERR116	007062	ERR60	004460	LPST3	001344
EROR12	002740	ERR117	007176	ERR61	004522	LPST4	001374
EROR14	003002	ERR12	001704	ERR62	004550	LPST40	003756
EROR15	003030	ERR120	007212	ERR63	004612	LPST5	001424
EROR16	003072	ERR121	007326	ERR64	004722	LPST56	004630
EROR17	003174	ERR122	007430	ERR65	004762	LPST6	001454
EROR2	002376	ERR123	007476	ERR66	005074	LPST7	001504
EROR20	003250	ERR124	007564	ERR67	005126	LPST74	005144
EROR21	003302	ERR125	007674	ERR7	001472	LPST77	005330
EROR23	003344	ERR126	007742	ERR70	005214	LPSX10	005664

LPX105	006234	STAI15	002046	TDMA	001060
LPX106	006326	STAI16	002116	TEXBUF	011044
LTSWT	005052	STAI17	002166	THREST	007530
MA	001032	STAI2	001270	TKB	001006
MES5	011073	STAI22	002242	TKS	001012
MES6	011116	STAI23	002336	TKSEL	010260
N	000127	STAI24	002412	TPB	001004
NOFMT	007710	STAI26	002452	TPS	001010
NXDST	005506	STAI27	002514	TRACK	001046
NXMTSM	006332	STAI3	001320	TSTCH	011046
OUTBUF	011144	STAI30	002542	TSTNEM	006442
PASS	001072	STAI31	002604	TSTRES	007452
PATNU	001052	STAI32	002652	TWRDCT	001056
RANNU	001042	STAI33	002716	TYEXIT	010676
RDFMT	007614	STAI35	002756	TYP	010576
RDWAT	004062	STAI36	003020	TYPA	010606
RDYINT	007512	STAI37	003046	TYPC	010624
RDYRES	007422	STAI4	001350	TYPD	010652
READ	104405	STAI40	003114	TYPDAT	010726
RESTOR	007356	STAI41	003210	TYPF	010700
RKDS	001026	STAI42	003264	TYPG	010712
RKER	001024	STAI44	003320	TYPS	010730
SAV1	001070	STAI45	003362	TYPSA	010754
SAVE	001066	STAI46	003410	TYPSB	010756
SCOPE	000004	STAI47	003452	VECTOR	001034
SELCMA	010170	STAI5	001400	WAFBUS	005404
SELDAR	010200	STAI50	003520	WC	001016
SELDCS	010210	STAI51	003564	WCBUSI	003524
SELWC	010160	STAI53	003624	WCWAT	003214
SR	001000	STAI54	003666	WRCHEC	104407
SRCHG	010412	STAI55	003714	WRD	001074
ST103X	005670	STAI56	003762	WRD1	001076
ST104X	006036	STAI57	004056	WRD2	001100
ST105X	006204	STAI6	001430	WRDCT	001044
ST106X	006264	STAI60	004132	WRITE	104403
STA100	005230	STAI62	004172	X103LP	006032
STA101	005274	STAI63	004234	X104LP	006200
STA103	005334	STAI64	004262	XSTAI1	001244
STA104	005502	STAI65	004324		
STA105	005636	STAI66	004372		
STA106	006632	STAI67	004436		
STA110	006762	STAI7	001460		
STA112	007112	STAI71	004476		
STA114	007242	STAI72	004540		
STAER	010424	STAI73	004566		
STAER1	010476	STAI74	005002		
STAI1	001212	STAI75	005110		
STAI10	001510	STAI77	005150		
STAI11	001566	START	001102		
STAI12	001652	STATUS	001036		
STAI13	001722	STX104	005442		
STAI14	001776	SWRDCT	001062		

```
ERRORS DETECTED: 0
RUN-TIME:  16 SECONDS
5K CORE USED
```

```
                000001                  BIT0=1
                000002                  BIT1=2
                000004                  BIT2=4
                000010                  BIT3=10
                000020                  BIT4=20
                000040                  BIT5=40
                000100                  BIT6=100
                000200                  BIT7=200
                000400                  BIT8=400
                001000                  BIT9=1000
                002000                  BIT10=2000
                004000                  BIT11=4000
                010000                  BIT12=10000
                020000                  BIT13=20000
                040000                  BIT14=40000
                100000                  BIT15=100000
                                        ;
                                        ;
                                        ;
                104403                  WRITE=TRAP+3
                104407                  WRCHECK=TRAP+7
                104405                  READ=TRAP+5
                104413                  RDCK=TRAP+13

                000000                  .=0
                000100                  .REPT   100
                                        .+2
                                        HALT
                                        .ENDR
000000  000002                          .+2
000002  000000                          HALT
000004  000006                          .+2
000006  000000                          HALT
000010  000012                          .+2
000012  000000                          HALT
000014  000016                          .+2
000016  000000                          HALT
000020  000022                          .+2
000022  000000                          HALT
000024  000026                          .+2
000026  000000                          HALT
000030  000032                          .+2
000032  000000                          HALT
000034  000036                          .+2
000036  000000                          HALT
000040  000042                          .+2
000042  000000                          HALT
000044  000046                          .+2
000046  000000                          HALT
000050  000052                          .+2
000052  000000                          HALT
000054  000056                          .+2
000056  000000                          HALT
000060  000062                          .+2
```

```
000062  000000                    HALT
000064  000066                    .+2
000066  000000                    HALT
000070  000072                    .+2
000072  000000                    HALT
000074  000076                    .+2
000076  000000                    HALT
000100  000102                    .+2
000102  000000                    HALT
000104  000106                    .+2
000106  000000                    HALT
000110  000112                    .+2
000112  000000                    HALT
000114  000116                    .+2
000116  000000                    HALT
000120  000122                    .+2
000122  000000                    HALT
000124  000126                    .+2
000126  000000                    HALT
000130  000132                    .+2
000132  000000                    HALT
000134  000136                    .+2
000136  000000                    HALT
000140  000142                    .+2
000142  000000                    HALT
000144  000146                    .+2
000146  000000                    HALT
000150  000152                    .+2
000152  000000                    HALT
000154  000156                    .+2
000156  000000                    HALT
000160  000162                    .+2
000162  000000                    HALT
000164  000166                    .+2
000166  000000                    HALT
000170  000172                    .+2
000172  000000                    HALT
000174  000176                    .+2
000176  000000                    HALT
000200  000202                    .+2
000202  000000                    HALT
000204  000206                    .+2
000206  000000                    HALT
000210  000212                    .+2
000212  000000                    HALT
000214  000216                    .+2
000216  000000                    HALT
000220  000222                    .+2
000222  000000                    HALT
000224  000226                    .+2
000226  000000                    HALT
000230  000232                    .+2
000232  000000                    HALT
000234  000236                    .+2
000236  000000                    HALT
```

```
000240  000242                    .+2
000242  000000                    HALT
000244  000246                    .+2
000246  000000                    HALT
000250  000252                    .+2
000252  000000                    HALT
000254  000256                    .+2
000256  000000                    HALT
000260  000262                    .+2
000262  000000                    HALT
000264  000266                    .+2
000266  000000                    HALT
000270  000272                    .+2
000272  000000                    HALT
000274  000276                    .+2
000276  000000                    HALT
000300  000302                    .+2
000302  000000                    HALT
000304  000306                    .+2
000306  000000                    HALT
000310  000312                    .+2
000312  000000                    HALT
000314  000316                    .+2
000316  000000                    HALT
000320  000322                    .+2
000322  000000                    HALT
000324  000326                    .+2
000326  000000                    HALT
000330  000332                    .+2
000332  000000                    HALT
000334  000336                    .+2
000336  000000                    HALT
000340  000342                    .+2
000342  000000                    HALT
000344  000346                    .+2
000346  000000                    HALT
000350  000352                    .+2
000352  000000                    HALT
000354  000356                    .+2
000356  000000                    HALT
000360  000362                    .+2
000362  000000                    HALT
000364  000366                    .+2
000366  000000                    HALT
000370  000372                    .+2
000372  000000                    HALT
000374  000376                    .+2
000376  000000                    HALT

        000200                    .=200
000200  000167  000712            JMP     START

                                  ;STATIC ROUTINES
        000300                            .=300
```

```
        000600                          .=600
000600  000167  001400                  JMP     ADT1            ;WRITE EACH SECTOR ON ITSELF AND
                                                                ;READ IT BACK TO VERIFY IT
                                                                ;
000604  000167  002024                  JMP     ADT2            ;TEST THE ABILITY ON THE DRIVE TO SEEK
                                                                ;FROM CYLINDER TO ALL OTHER CYLINDERS
                                                                ;
000610  000167  002454                  JMP     ADT2UX          ;TEST THE CYLINDER SELECTION MATRIX
                                                                ;
000614  000167  003154                  JMP     ADT3UX          ;THIS TEST IS A MAX. HEAD MOVEMENT TEST.
                                                                ;THE TEST SEEKS FROM 0 TO 312 THEN
                                                                ;FROM 312 TO 0 THEN IT DEC. THE HIGH
                                                                ;COUNT AND INC. THE LOW COUNT AND
                                                                ;SEEKS TIL 0 BECOMES 312 AND HIGH BECOMES 0
                                                                ;
000620  000167  003552                  JMP     SAWTH           ;IN THIS TEST THE HEADS APPEAR TO
                                                                ;BE MOVING IN A SAW TOOTH PATTERN
                                                                ;WHICH IS TO GENERATE WORST CASE NOISE
                                                                ;
000624  000167  004220                  JMP     ADT3            ;THIS TEST CHECKS THE LOOK AHEAD FEATURE
                                                                ;OF THE CONTROL
                                                                ;
000630  000167  005162                  JMP     RANEX           ;RANDOM ADDRESS, DATA AND
                                                                ;WORD COUNT TEST
                                                                ;
                                        ;***MAINTENANCE ROUTINES***
                                        ;
000634  000167  010670                  JMP     SELWC           ;LOAD WORD COUNT REG.
                                                                ;WITH SWITCH REGISTER
                                                                ;
000640  000167  010674                  JMP     SELCMA          ;LOAD CURRENT MEMORY ADDR
                                                                ;REG. WITH SWITCH REGISTER
                                                                ;
000644  000167  010700                  JMP     SELDAR          ;LOAD DISK ADDR. REGISTER
                                                                ;WITH SWITCH REGISTER
                                                                ;
000650  000167  010704                  JMP     SELDCS          ;LOAD DISK CONTROL REGISTER
                                                                ;WITH SWITCH REGISTER
                                                                ;
000654  000167  011562                  JMP     TKSEL           ;ENABLE READ AMPLIFIERS
                                                                ;TO TRACK SELECTED
                                                                ;FROM SWITCH REGISTER
                                                                ;
```

```
                                                        ;
000660  000167  010744          JMP     PFT1            ;POWER FAIL TEST 1WRITE DRIVE WHEN
                                                        ;FOR LOSS OF DATA
                                                        ;LOSSING POWER THEN GO BACK AND READ IT
                                                        ;
                                                        ;
000664  000167  011126          JMP     PFT2            ;POWER FAIL TEST 2 READ DRIVE WHEN
                                                        ;LOSSING POWER THEN GO BACK AND CHECK IT
                                                        ;FOR LOSS OF DATA
                                                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;RK11 DATA TEST
                        ;VECTORS USED IN PROGRAM
                        ;#1 LOC 204 DISK INTERRUPT
                        ;#2 LOC 30 EMT (TELETYPE OUTPUT)
                        ;#3 LOC 34 TRAP (DISK HANDLERS)
                        ;#4 LOC 14 TRACE TRAP (USED IN BACKGROUND TEST)
                        ;#5 LOC 20 IOT TRAP (USED IN CALLING BACKGROUND TEST)

        001000          .=1000
```

```
001000                          .EVEN
                                ;
                                ;I/O ADDRESS POINTERS
001000  177570                  SR:     177570                  ;SWITCH REGIISTER
001002  177776                  CSR:    177776                  ;PROCESSOR STATUS REGISTER
001004  177566                  TPB:    177566                  ;TELETYPE REGISTERS
001006  177562                  TKB:    177562
001010  177564                  TPS:    177564
001012  177560                  TKS:    177560
                                ;
                                ;DISK I/O REGISTERS
                                ;
001014  177404                  RKCS:   177404                  ;DISK CONTROL REGISTER
001016  177406                  RKWC:   177406                  ;WORD COUNT REGISTER
001020  177410                  RKBA:   177410                  ;CURRENT ADDRESS REGISTER
001022  177412                  RKDA:   177412                   ;DISK ADDRESS REGISTER
001024  177402                  RKER:   177402                  ;ERROR REGISTER
001026  177400                  RKDS:   177400                  ;DRIVE STATUS REGISTER
001030  177416                  RKDB:   177416                  ;DATA BUFFER REGISTER
001032  177414                  RKMR:   177414                  ;MAINTENANCE REGISTER
001034  000220                  VECTOR: 220                     ;INTERRUPT VECTOR ADDRESS
001036  000222                  STATUS: 222                     ;DISK INTERRUPT STATUS
001040  000200                  PRIORITY:BIT7                   ;DISK PRIORITY LEVEL

                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;RK11 DEDICATE REGISTERS (MEMORY)
                                ;
001042  000000                  FLAG:   0                       ;INTERNAL PROGRAM FLAG
001044  146723                  RANNU:  146723                  ;RANDOM NUMBER PRIME
001046  000000                  WRDCT:  0                       ;WORKING WORD COUNT
001050  000000                  TRACK:  0                       ;WORKING DAE
001052  000000                  DMA:    0                       ;WORKING DAR
001054  000000                  PATNU:  0                       ;DATA PATTERN INDEX
001056  000000                  BUF:    0                       ;WORKING DATA BUFFER (OUT-IN)
001060  000000                  TWRDCT: 0                       ;TEMP WORD COUNT
001062  000000                  TDMA:   0                       ;TEMP DAR
001064  000000                  SWRDCT: 0                       ;STANDARD WORD COUNT
001066  000000                  ERCOUNT:0                       ;ERROR COUNT FOR MESSAGES.
001070  000000                  SAVE:   0
001072  000000                  SAV1:   0
001074  000000                  PASS:   0
001076  000000                  DSKNOR: 0
001100  000000                  HRDER:  0                       ;POINTER FOR HARD ERROR
001102  000000                  BLOCK:  0
001104  000000                  PASSC:  0
001106  000000                  INBUF:  0

                                ;
                                ;RK11 WORK REGISTERS
```

```
                                ;(CAN BE CHANGED IN ANY ROUTINE)
001110  000000                  WORK:   0
001112  000000                  WORK1:  0
001114  000000                  WORK2:  0
```

```
001116  000005                  START:  RESET                   ;CLEAR THE WORLD
001120  012706  001000                  MOV     #1000,%6        ;SET UP STACK
001124  004567  010110                  JSR     %5,EXTMEN       ;SET UP DATA BUFFERS
001130  012706  001000                  MOV     #1000,%6        ;RESET STACK
001134  012767  012722  176666          MOV     #EMTRP,30       ;SET UP TTY POINTER
001142  012767  000340  176662          MOV     #340,32         ;LOCK UP INTERRUPTS
001150  012767  006514  176656          MOV     #DISK,34        ;SET UP DISK HANDLER POINTER
001156  012767  000340  176652          MOV     #340,36         ;LOCK UP INTERRRUPTS
001164  012777  000340  177610          MOV     #340,@CSR       ;LOCK UP INTERRUPT LEVELS
001172  012767  011440  176620          MOV     #XWAIT,20
001200  005067  176616                  CLR     22
001204  005067  177632                  CLR     FLAG            ;CLEAR PROGRAM FLAG
001210  005067  177636                  CLR     DMA             ;CLEAR DAR REGISTERS
001214  005067  177634                  CLR     PATNU           ;CLEAR PATTER COUNT
001220  016767  177640  177620          MOV     SWRDCT,WRDCT
001226  005777  177546                  TST     @SR
001232  100432                          BMI     LCONM           ;OPERATE UNDER PROGRAM CONTROL
001234  052767  070100  177600          BIS     #70100,FLAG
001242  104001                  SPECOR:   EMT   +1
001244  014107                          CON3
001246  004767  011730                  JSR     %7,NOCHA        ;HOW MANY DRIVES
001252  162767  000001  012132          SUB     #1,TEXBUF
001260  022767  000010  012124          CMP     #10,TEXBUF      ;TEST FOR MAX NUMBER OF DRIVES
001266  101765                          BLOS    SPECOR          ;NUMBER TOO LARGE
001270  016767  012116  177600          MOV     TEXBUF,DSKNOR
001276  006167  177574                  ROL     DSKNOR
001302  006167  177570                  ROL     DSKNOR
001306  052767  004000  177526          BIS     #BIT11,FLAG     ;SET UP PROGRAM FLAG
001314  000167  000572                  JMP     ADTST
                        ;ENTER OPERATOR CONVERSATION MODE
001320  004567  011574          LCONM:  JSR     %5,     CONV    ;CONVERT SWRDCT TO ASCII
001324  001064                          SWRDCT
001326  014004                          SPECMES
001330  000006                          6
001332  104000                          EMT+0
001334  013500                          HED5A
001336  014004                          SPECMES
001340  177777                          -1
001342  104001                  CONM:   EMT     +1              ;ASK ABOUT DATA TEST ONLY
001344  014022                          CON1
001346  004767  011640                  JSR     %7,ALPHA        ;GO WAIT FOR ANSWER
001352  022767  000153  012032          CMP     #153,TEXBUF     ;TEST FOR YES
001360  001003                          BNE     .+10            ;BRANCH IF NO
001362  052767  002000  177452          BIS     #BIT10,FLAG
001370  104001                          EMT     +1
001372  014054                          CON2                    ;ASK ABOUT MULTI DRIVE MODE
001374  004767  011612                  JSR     %7,ALPHA        ;GO WAIT FOR ANSWER
001400  022767  000153  012004          CMP     #153,TEXBUF     ;TEST FOR YES
001406  001026                          BNE     DATTES          ;
001410  052767  004000  177424          BIS     #BIT11,FLAG     ;
001416  104001                  DSKDR:  EMT     +1
001420  014107                          CON3
001422  004767  011554                  JSR     %7,NOCHA
001426  162767  000001  011756          SUB     #1,TEXBUF
```

```
001434  022767  000010  011750          CMP     #10,TEXBUF
001442  101765                          BLOS    DSKDR
001444  016767  011742  177424          MOV     TEXBUF,DSKNOR
001452  006167  177420                  ROL     DSKNOR
001456  006167  177414                  ROL     DSKNOR
001462  000420                          BR      ASKWC
001464  104001                  DATTES: EMT     +1
001466  014150                          CON4                    ;ASK UNIT NUMBER
001470  004767  011506                  JSR     %7,NOCHA        ;WAIT FOR NO.
001474  022767  000010  011710          CMP     #10,TEXBUF      ;IS NO = OR>10
001502  101770                          BLOS    DATTES          ;NO
001504  000241                          CLC
001506  006167  011700                  ROL     TEXBUF
001512  006167  011674                  ROL     TEXBUF
001516  056767  011670  177316          BIS     TEXBUF,FLAG
001524  104001                  ASKWC:  EMT     +1
001526  014171                          CON5                    ;ASK ABOUT OPTIONAL WORD COUNT
001530  004767  011456                  JSR     %7,ALPHA        ;WAIT FOR ANSWER
001534  022767  000153  011650          CMP     #153,TEXBUF
001542  001026                          BNE     TKSR            ;ASK ABOUT OPTIONAL DAR
001544  104001                  WCCON:  EMT     +1
001546  014227                          CON6                    ;ASK LENGTH OF WC
001550  004767  011426                  JSR     %7,NOCHA
001554  005767  011632                  TST     TEXBUF
001560  001771                          BEQ     WCCON
001562  016767  177276  177320          MOV     SWRDCT,WORK
001570  005267  177314                  INC     WORK
001574  026767  177310  011610          CMP     WORK,TEXBUF     ;IS TEXBUF LESSTHAN SWRDCT
001602  101760                          BLOS    WCCON           ;YES ASK FOR COUNT AGAIN
001604  016767  011602  177252          MOV     TEXBUF,SWRDCT   ;OPERATING WORD COUNT
001612  016767  177246  177226          MOV     SWRDCT,WRDCT
001620  104001                  TKSR:   EMT+1                   ;ASK ABOUT CYCINDER #
001622  014314                          CON7A
001624  004767  011362                  JSR     %7,ALPHA
001630  022767  000153  011554          CMP     #153,TEXBUF
001636  001003                          BNE     OPDAR
001640  052767  000040  177174          BIS     #BIT5,FLAG
001646  104001                  OPDAR:  EMT     +1
001650  014256                          CON7                    ;ASK ABOUT OPTIONAL DAR
001652  004767  011324                  JSR     %7,NOCHA
001656  022767  000034  011526          CMP     #34,TEXBUF      ;33 MAX SECTOR COUNT
001664  101770                          BLOS    OPDAR
001666  016767  011520  177156          MOV     TEXBUF, DMA     ;TEMP SECTOR REGISTER
001674  042767  177740  177150          BIC     #177740,DMA
```

```
001702  104001                  OPPAT:  EMT     +1
001704  014362                          CON8                    ;ASK ABOUT DATA PATTERNS
001706  004767  011270                  JSR     %7,NOCHA
001712  022767  000023  011472          CMP     #23,TEXBUF      ;TEST FOR CORRECT NO
001720  101770                          BLOS    OPPAT           ;ASK AGAIN
001722  022767  000022  011462          CMP     #22,TEXBUF
001730  001414                          BEQ     OPWRT           ;DATA PATTERN UNDER PROGRAM CONTROL
001732  052767  100000  177102          BIS     #BIT15,FLAG     ;SET PROGRAM FLAG
001740  016767  011446  177106          MOV     TEXBUF, PATNU   ;OPERATOR WANTS TO SELECT DATA
001746  000241                          CLC
001750  006167  177100                  ROL     PATNU
001754  042767  070100  177060          BIC     #70100,FLAG     ;CLEAR OP MODE BITS IN FLAG
001762  104001                  OPWRT:  EMT     +1
001764  014416                          CON9                    ;ASK ABOUT WRITE
001766  004767  011220                  JSR     %7,ALPHA
001772  022767  000153  011412          CMP     #153,TEXBUF     ;TEST FOR YES
002000  001003                          BNE     OPRD            ;ASK ABOUT READ
002002  052767  040000  177032          BIS     #BIT14,FLAG     ;YES SET FLAG BIT
002010  104001                  OPRD:   EMT     +1
002012  014464                          CON11                   ;ASK ABOUT READ
002014  004767  011172                  JSR     %7,ALPHA
002020  022767  000153  011364          CMP     #153,TEXBUF     ;TEST FOR YES ANSWER
002026  001003                          BNE     RDCHK
002030  052767  010000  177004          BIS     #BIT12,FLAG     ;SET FLAG TO READ
002036  104001                  RDCHK:  EMT+1
002040  014503                          CON12
002042  004767  011144                  JSR     %7,ALPHA
002046  022767  000153  011336          CMP     #153,TEXBUF
002054  001003                          BNE     OPWCK
002056  052767  000100  176756          BIS     #BIT6,FLAG
002064  104001                  OPWCK:  EMT+1
002066  014436                          CON10                   ;ASK ABOUT WRITE CHECK
002070  004767  011116                  JSR     %7,ALPHA
002074  022767  000153  011310          CMP     #153,TEXBUF
002102  001003                          BNE     ADTST
002104  052767  020000  176730          BIS     #BIT13,FLAG
                                ;
                                ;
                                ;
002112  032767  004000  176722  ADTST:  BIT     #BIT11,FLAG     ;ARE WE IN MULTI DRIVE MODE
002120  001423                          BEQ     EXMFLG          ;BRANCH IF NO.
002122  104001                          EMT     +1
002124  013731                          MES11
002126  016767  176710  011044          MOV     FLAG,ACNVX
002134  006067  011040                  ROR     ACNVX
002140  006067  011034                  ROR     ACNVX
002144  042767  177770  011026          BIC     #177770,ACNVX   ;FETCH DRIVE #
002152  004567  010742                  JSR     %5,CONV
002156  013200                          ACNVX
002160  013743                          MES12
002162  000001                          1
002164  104001                          EMT     +1
002166  013743                          MES12
002170  032767  002000  176644  EXMFLG: BIT     #BIT10,FLAG     ;TEST FOR DATA TEST ONLY
```

```
002176  001402                  BEQ     .+6             ;DO COMPLETE TEST
002200  000167  003116           JMP     DATAT           ;DO DATA TEST ONLY
```

```
                                ;
                                ;***** ADDRESS TEST *****
                                ;
                                ;
                                ;RK11 ADDRESS TEST #1 (TRACK AND SECTOR SELECTION TEST)
                                ;
                                ;WRITE 200(OCTAL) WORDS IN EACH SECTOR
                                ;THE WORD CONTAINS THE ADDRESS OF EACH SECTOR
                                ;WHEN THE COMPLETE DISK IS WRITTEN READ
                                ;BACK EACH SECTOR AND COMPARE FOR THE CORRECT
                                ;DATA IN THE SECTOR
002204  016767  176654  176656  ADT1: MOV     SWRDCT,SAVE
002212  004567  005744            JSR     %5,DSKNOS       ;SET UP DISK NUMBER
002216  056777  176666  176576    BIS     WORK,@RKDA      ;SET UP DRIVE NUMBER
002224  012777  000015  176562    MOV     #15,@RKCS       ;RESET DRIVE
002232  105777  176556            TSTB    @RKCS           ;WAIT FOR DRIVE READY
002236  100375                    BPL     .-4
002240  032777  000100  176560    BIT     #BIT6,@RKDS
002246  001774                    BEQ     .-6
002250  012767  000200  176606    MOV     #200,SWRDCT
002256  012767  000200  176562    MOV     #200,WRDCT      ;SETUP WORD COUNT
002264  012767  014536  176564    MOV     #OUTBUF,BUF     ;SET UP CURRENT ADDRESS
002272  012706  001000            MOV     #1000,%6        ;SET UP STACK
002276  005067  176550            CLR     DMA
002302  012777  000001  176504  SEABUF: MOV     #BIT0,@RKCS
002310  105777  176500            TSTB    @RKCS
002314  100375                    BPL     .-4
002316  012700  014536            MOV     #OUTBUF,%0      ;SET UP ADDRESS BUFFER
002322  016701  176520            MOV     WRDCT,%1        ;
002326  016720  176520  XSEABUF: MOV     DMA,(0)+
002332  005301                    DEC     %1
002334  001374                    BNE     XSEABUF
002336  104403                    WRITE                   ;WRITE SECTOR
002340  105777  176450            TSTB    @RKCS           ;CHECK FOR READY
002344  100375                    BPL     .-4
002346  005777  176442            TST     @RKCS           ;TEST FOR ERROR
002352  100014                    BPL     WRNEXB          ;BRANCH IF NO ERROR
002354  012767  000001  176504    MOV     #1,ERCOUNT
002362  017767  176426  176520    MOV     @RKCS,WORK      ;FETCH CONTENTS OF CONTROL REG
002370  017767  176432  176514    MOV     @RKDS,WORK1     ;FETCH CONTENTS OF ADDRESS REG
002376  004567  010244  ERR1:    JSR     %5,STAER1       ;REPORT ERROR
002402  000737                    BR      SEABUF          ;LOOP ON ERROR
002404  004767  005054  WRNEXB: JSR     %7,DISBUF       ;SET UP NEXT DISK ADDR.
002410  000734                    BR      SEABUF          ;WRITE NEXT SECTOR
002412  005067  176434            CLR     DMA
002416  016767  176464  176432  SETADT: MOV     INBUF,BUF       ;SET UP CURRENT ADDRESS
002424  012777  000015  176362    MOV     #15,@RKCS       ;RESET DRIVE
002432  105777  176356            TSTB    @RKCS
002436  100375                    BPL     .-4
002440  032777  000100  176360    BIT     #BIT6,@RKDS
002446  001774                    BEQ     .-6
```

```
002450  012777  000001  176336  RDSECT: MOV     #BIT0,@RKCS
002456  105777  176332          TSTB    @RKCS
002462  100375                  BPL     .-4
002464  104405                  READ
002466  105777  176322          TSTB    @RKCS           ;CHECK FOR READY
002472  100375                  BPL     .-4             ;NOT READY BRANCH BACK
002474  005777  176314          TST     @RKCS           ;TEST FOR ERROR
002500  100014                  BPL     ADHGT           ;BRANCH IF NO ERROR
002502  017767  176306  176400  MOV     @RKCS,WORK      ;FETCH CONTROL REG
002510  017767  176306  176374  MOV     @RKDA,WORK1     ;FETCH DISK ADDR
002516  012767  000002  176342  MOV     #2,ERCOUNT      ;SET UP ERROR COUNT
002524  004567  010116          ERR2:   JSR     %5,STAER1       ;REPORT ERROR
002530  000747                  BR      RDSECT          ;LOOP ON ERROR
002532  016700  176350          ADHGT:  MOV     INBUF,%0
002536  012701  000200          MOV     #200,%1
002542  026710  176304          SANHT:  CMP     DMA,(0)         ;CMP FOR CORRECT ADDR.
002546  001004                  BNE     ADERR           ;BRANCH IF DATA DID NOT COMPARE
002550  005720                  INCADT: TST     (0)+
002552  005301                  DEC     %1
002554  001372                  BNE     SANHT           ;TEST NEXT WORD
002556  000412                  BR      CHKADT
002560  016767  176266  176324  ADERR:  MOV     DMA,WORK1       ;CORRECT ADDRESS
002566  011067  176316          MOV     (0),WORK        ;DATA IN ERROR
002572  012767  000003  176266  MOV     #3,ERCOUNT      ;SET UP ERROR COUNT
002600  004567  010042          ERR3:   JSR     %5,STAER1       ;REPORT ERROR
                                ;*****REPORT ONLY ONE ERROR PER SECTOR*****
002604  004767  004654          CHKADT: JSR     %7,DISBUF       ;SET UP NEXT DISK BUFFER
002610  000717                  BR      RDSECT          ;CHECK NEXT SECTOR
002612  016767  176252  176244  MOV     SAVE,SWRDCT     ;RESTORE STANDARD WORD COUNT
002620  032777  004000  176152  BIT     #BIT11,@SR      ;DOES OPERATOR WANT TO LOOP ON TEST
002626  001402                  BEQ     .+6
002630  000167  177350          JMP     ADT1
```

```
                                ;ADDRESS TEST #2
                                ;POLE TEST
                                ;
                                ;SEEK FROM CYLINDER 0 TO ALL OTHER CYLINDERS
                                ;
002634  005067  176212          ADT2:   CLR     DMA
002640  012777  000001  176146          MOV     #BIT0,@RKCS     ;RESET CONTROL
002646  105777  176142                  TSTB    @RKCS
002652  100375                          BPL     .-4
002654  012706  001000                  MOV     #1000,%6        ;SET UP STACK
002660  012777  005022  176146          MOV     #PWAIT,@VECTOR  ;SET UP INTERRUPT VECTOR
002666  012777  000340  176142          MOV     #340,@STATUS
002674  016777  176140  176100          MOV     PRIORITY,@CSR
002702  005077  176114                  CLR     @RKDA
002706  004567  005250                  JSR     %5,DSKNOS       ;SET UP THE DISK
002712  056777  176172  176102          BIS     WORK,@RKDA      ;SET UP THE DRIVE NUMBER
002720  012777  000011  176066          MOV     #11,@RKCS       ;SEEK FOR ZERO
002726  105777  176062                  TSTB    @RKCS           ;TEST FOR CONTROL READY
002732  100375                          BPL     .-4
002734  032777  000100  176064          BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
002742  001774                          BEQ     .-6
002744  056777  176102  176050          BIS     DMA,@RKDA
002752  012777  000011  176034          MOV     #11,@RKCS       ;GO SEEK CYLINDER
                                ;****WAIT FOR CONTROL READY IN FLAG MODE****
002760  005067  176124                  CLR     WORK
002764  105777  176024          TSRDY:  TSTB    @RKCS           ;TEST FOR CONTROL READY
002770  100422                          BMI     ADT2X           ;BRANCH IF CONTROL READY
002772  005267  176112                  INC     WORK            ;+1 COUNTER FOR READY
002776  001372                          BNE     TSRDY           ;TEST FOR READY AGAIN
003000  012777  000340  175774          MOV     #340,@CSR       ;LOCKUP CPU
003006  012767  000004  176052          MOV     #4,ERCOUNT
003014  017767  175774  176066          MOV     @RKCS,WORK      ;FETCH CONTROL STATUS REG
003022  017767  176000  176062          MOV     @RKDS,WORK1     ;FETCH DRIVE STATUS REG
003030  004567  007612          ERR4:   JSR     %5,STAER1       ;REPORT ERROR
003034  000677                          BR      ADT2            ;LOOP ON ERROR
                                ;*****WAIT FOR INTERRUPT FROM DRIVE READY*****
003036  005067  176046          ADT2X:  CLR     WORK
003042  052777  000100  175744          BIS     #BIT6,@RKCS     ;SET INTERRUPT ENABLE
003050  000777                          BR      .               ;WAIT FOR INTERRUPT WITH BRANCH .
003052  032777  000100  175746          BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003060  001017                          BNE     LPAD2           ;DRIVE READY OK!
003062  012777  000340  175712          MOV     #340,@CSR       ;LOCK UP CPU
003070  012767  000005  175770          MOV     #5,ERCOUNT
003076  017767  175712  176004          MOV     @RKCS,WORK      ;FETCH CONTROL REG
003104  017767  175716  176000          MOV     @RKDS,WORK1     ;FETCH DRIVE STATUS
003112  004567  007530          ERR5:   JSR     %5,STAER1
003116  000646                          BR      ADT2            ;LOOP ON ERROR
003120  017767  175702  175762  LPAD2:  MOV     @RKDS,WORK      ;FETCH DRIVE STATUS
003126  000241                          CLC
003130  006167  175754                  ROL     WORK
003134  006167  175750                  ROL     WORK
003140  006167  175744                  ROL     WORK
003144  006167  175740                  ROL     WORK
003150  042767  177770  175732          BIC     #177770,WORK
003156  016767  175660  175726          MOV     FLAG,WORK1
```

```
003164  006067  175722                    ROR     WORK1
003170  006067  175716                    ROR     WORK1
003174  042767  177770  175710            BIC     #177770,WORK1
003202  026767  175704  175700            CMP     WORK1,WORK      ;IS THE POLE DRIVE NUMBER CORRECT
003210  001407                            BEQ     POLOK           ;YES DRIVE NUMBER OK
003212  012767  000006  175646  ERR6:     MOV     #6,ERCOUNT      ;SET UP ERROR COUNT
003220  004567  007422                    JSR     %5,STAER1       ;REPORT NUMBER INCORRECT
003224  000167  177404                    JMP     ADT2            ;LOOP ON TEST
003230  022767  014500  175614  POLOK:CMP       #14500,DMA
003236  001404                            BEQ     .+12
003240  062767  000040  175604            ADD     #40,DMA
003246  000406                            BR      EXTAD2
003250  032777  004000  175522            BIT     #BIT11,@SR      ;CHECK TO LOOP ON TEST
003256  001404                            BEQ     ADT2UX          ;GO TO NEXT TEST
003260  000167  177350                    JMP     ADT2            ;LOOP ON TEST
003264  000167  177350          EXTAD2:   JMP     ADT2+4
```

```
                        ;
                        ;POLE TEST WITH ADDRESS VERIFICATION
                        ;
                        ;
                        ;WRITE THE FIRST WORD IN SECTOR 0
                        ;OF EACH CYLINDER WITH ITS SELF
                        ;
                        ;THEN POLE EACH CYLINDER AND CHECK
                        ;THE ADDRESS

003270  005067  175556        ADT2UX: CLR     DMA
003274  012777  000001  175512        MOV     #BIT0,@RKCS     ;CLEAR CONTROL
003302  105777  175506        WRAUSK: TSTB    @RKCS
003306  100375                        BPL     .-4
003310  012706  001000                MOV     #1000,%6        ;SET UP STACK
003314  012777  005022  175512        MOV     #PWAIT,@VECTOR  ;SET UP INTERRUPT VECTOR
003322  012777  000340  175506        MOV     #340,@STATUS
003330  016777  175504  175444        MOV     PRIORITY,@CSR
003336  012767  000001  175502        MOV     #1,WRDCT        ;SET UP WORD C
003344  012767  014536  175504        MOV     #OUTBUF,BUF
003352  016737  175474  014536        MOV     DMA,@#OUTBUF
003360  104403                        WRITE
003362  105777  175426                TSTB    @RKCS
003366  100375                        BPL     .-4
003370  026727  175456  014500        CMP     DMA,#14500      ;IS IT THE LAST CYLINDER
003376  001404                        BEQ     A2XSEK-4
003400  062767  000040  175444        ADD     #40,DMA
003406  000761                        BR      WRAUSK
003410  005067  175436                CLR     DMA
003414  004567  004542        A2XSEK: JSR     %5,DSKNOS       ;SET UP DISK NO.
003420  056777  175464  175374        BIS     WORK,@RKDA      ;SET UP THE DRIVE
003426  042777  017777  175366        BIC     #17777,@RKDA
003434  052777  014500  175360        BIS     #14500,@RKDA    ;SET UP SEEK TO 300
003442  012777  000011  175344        MOV     #11,@RKCS       ;GO SEEK
003450  105777  175340                TSTB    @RKCS
003454  100375                        BPL     .-4
003456  032777  000100  175342        BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003464  001774                        BEQ     .-6
003466  042777  017777  175326        BIC     #17777,@RKDA
003474  056777  175352  175320        BIS     DMA,@RKDA
003502  012777  000011  175304        MOV     #11,@RKCS       ;GO SEEK CYLINDER
                        ;****WAIT FOR CONTROL READY IN FLAG MODE****
003510  005067  175374                CLR     WORK
003514  105777  175274        TSTAUX: TSTB    @RKCS           ;WAIT FOR CONTROL RDY
003520  100422                        BMI     AUX2            ;CONTROL READY OK
003522  005267  175362                INC     WORK
003526  001372                        BNE     TSTAUX          ;GO WAIT FOR CONTROL RDY
003530  012777  000340  175244        MOV     #340,@CSR
003536  012767  000007  175322  ERR7: MOV     #7,ERCOUNT
003544  017767  175244  175336        MOV     @RKCS,WORK      ;REPORT RKCS
003552  017767  175250  175332        MOV     @RKDS,WORK1     ;REPORT NO CONTROL READY
003560  004567  007022                JSR     %5,STAER        ;REPORT ERROR
003564  000713                        BR      A2XSEK
                        ;****WAIT FOR INTERRUPT FROM DRIVE READY****
```

```
003566  005067  175316          AUX2:  CLR     WORK
003572  052777  000100  175214         BIS     #BIT6,@RKCS     ;SET INTERRUPT ENABLE
003600  000777                         BR      .
003602  032777  000100  175216         BIT     #BIT6,@RKDS     ;TEST FOR DRIVE READY
003610  001017                         BNE     AU2LPA          ;DRIVE READY OK
```

```
003612  012777  000340  175162          MOV     #340,@CSR
003620  012767  000010  175240  ERR10:  MOV     #10,ERCOUNT
003626  017767  175162  175254          MOV     @RKCS,WORK
003634  017767  175162  175250          MOV     @RKDA,WORK1
003642  004567  007000                  JSR     %5,STAER1       ;REPORT SEEK COMPLETE DRIVE NOT READY
003646  000662                          BR      A2XSEK
003650  012777  177777  175140  AU2LPA: MOV     #-1,@RKWC
003656  016777  175224  175134          MOV     INBUF,@RKBA
003664  012777  000005  175122          MOV     #5,@RKCS        ;GO READ SECTOR
003672  105777  175116                  TSTB    @RKCS           ;CHECK FOR READY
003676  100375                          BPL     .-4
003700  026777  175146  175200          CMP     DMA,@INBUF      ;IS DATA CORRECT
003706  001414                          BEQ     LPAU2X
003710  016767  175136  175174          MOV     DMA,WORK1       ;WHAT SECTOR SHOULD CONTAIN
003716  017767  175164  175164          MOV     @INBUF,WORK     ;WHAT WAS IN SECTOR
003724  012767  000011  175134  ERR11:  MOV     #11,ERCOUNT
003732  004567  006710                  JSR     %5,STAER1       ;REPORT ERROR
003736  000626                          BR      A2XSEK
003740  022767  014500  175104  LPAU2X: CMP     #14500,DMA      ;IS IT THE LAST CYLINDER
003746  001404                          BEQ     .+12            ;YES
003750  062767  000040  175074          ADD     #40,DMA         ;SET UP NEXT CYLINDER
003756  000616                          BR      A2XSEK          ;SEEK NEXT CYLINDER
003760  032777  004000  175012          BIT     #BIT11,@SR      ;LOOP ON TEST?
003766  001402                          BEQ     .+6
003770  000167  177274                  JMP     ADT2UX          ;LOOP ON TEST
                                ;
                                ;POLE TEST #3
                                ;SEEK FROM 0 - 312 THEN
                                ;SEEK FROM 312 - 0 THEN DECREMENT THE 312 COUNT
                                ;AND INCREMENT 0 THEN GO AND SEEK AGAIN
                                ;SEEK FROM 1 - 311 THEN
                                ;SEEK FROM 311 - 1
                                ;WHEN THE COUNTS HAVE GONE FROM 0 - 312
                                ;AND 312 - 0 THE TEST IS DONE.
                                ;
003774  005067  175070  ADT3UX: CLR     SAVE
004000  005067  175100          CLR     PASSC
004004  012767  014500  175060          MOV     #14500,SAV1
004012  012777  000001  174774          MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
004020  105777  174770                  TSTB    @RKCS           ;WAIT FOR CONTROL READY
004024  100375                          BPL     .-4
004026  012706  001000                  MOV     #1000,%6        ;SET UP STACK
004032  012777  005022  174774          MOV     #PWAIT,@VECTOR  ;SET UP INTERRUPT VECTOR
004040  012777  000340  174770          MOV     #340,@STATUS
004046  016777  174766  174726          MOV     PRIORITY,@CSR
004054  004567  004102          SEEKCY: JSR     %5,DSKNOS       ;SET UP DISK NUMBER
004060  056777  175024  174734          BIS     WORK,@RKDA      ;SET UP THE DRIVE NUMBER
004066  042777  017777  174726          BIC     #17777,@RKDA
004074  056777  174770  174720          BIS     SAVE,@RKDA      ;SET UP CYLINDER ADDRESS
004102  012777  000011  174704          MOV     #11,@RKCS       ;GO SEEK
                                ;****WAIT FOR READY IN FLAG MODE****
004110  005067  174774                  CLR     WORK
```

```
004114  105777  174674          TSTRA3: TSTB    @RKCS           ;TEST FOR READY
004120  100417                          BMI     NBOSL           ;READ SET OK
004122  005267  174762                  INC     WORK
004126  001372                          BNE     TSTRA3          ;GO WAIT FOR READY
004130  017767  174660  174754          MOV     @RKCS,WORK1
004136  016767  174726  174744          MOV     SAVE,WORK       ;CYLINDER #OF SEEK
004144  012767  000012  174714  ERR12:  MOV     #12,ERCOUNT
004152  004567  006470                  JSR     %5,STAER1       ;REPORT CONTROL NOT RDY
004156  000706                          BR      ADT3UX          ;LOOP ON TEST
                                ;****WAIT FOR DRIVE READY INTERRUPT****
004160  005067  174724          NBOSL:  CLR     WORK
004164  052777  000100  174622          BIS     #BIT6,@RKCS     ;5 CT INTERRUPT ENABLE
004172  000777                          BR      .
004174  004567  003762                  JSR     %5,DSKNOS
004200  056777  174704  174614          BIS     WORK,@RKDA      ;SET UP THE DRIVE NUMBER
004206  042777  017777  174606          BIC     #17777,@RKDA
004214  056777  174652  174600          BIS     SAV1,@RKDA
004222  012777  000011  174564          MOV     #11,@RKCS       ;GO SEEK
                                ;****WAIT FOR CONTROL READY IN FLAG MODE***
004230  005067  174654                  CLR     WORK
004234  105777  174554          CHKRA3:TSTB     @RKCS
004240  100417                          BMI     AISRY           ;READY SET OK
004242  005267  174642                  INC     WORK
004246  001372                          BNE     CHKRA3          ;GO WAIT FOR FLAG
004250  017767  174540  174634          MOV     @RKCS,WORK1
004256  016767  174610  174624          MOV     SAV1,WORK       ;CYLINDER # OF SEEK
004264  012767  000013  174574  ERR13:  MOV     #13,ERCOUNT     ;SET UP ERROR COUNT
004272  004567  006350                  JSR     %5,STAER1       ;REPORT ERROR
004276  000636                          BR      ADT3UX
                                ;****WAIT FOR SEEK COMPLETE IN INTERRUPT MODE***
004300  005067  174604          AISRY:  CLR     WORK
004304  052777  000100  174502          BIS     #BIT6,@RKCS     ;SET INTERRUPT ENABLE
004312  000777                          BR      .
                                ;
                                ;
                                ;
004314  005767  174564                  TST     PASSC           ;TEST FOR LAST PASS
004320  001003                          BNE     .+10
004322  012767  177770  174554          MOV     #-10,PASSC      ;SET UP NEW COUNT
004330  005267  174550                  INC     PASSC           ;+1 COUNT
004334  001011                          BNE     LPAU3X-4
004336  005767  174530                  TST     SAV1
004342  001410                          BEQ     LPAU3X
004344  062767  000040  174516          ADD     #40,SAVE        ;INC. LOW CYLINDER ADDR. REG.
004352  162767  000040  174512          SUB     #40,SAV1        ;DEC HI CYLINDER ADDR. REG.
004360  000167  177470                  JMP     SEEKCY          ;GO SEEK
004364  032777  004000  174406  LPAU3X: BIT     #BIT11,@SR      ;LOOP ON TEST
004372  001200                          BNE     ADT3UX          ;YES LOOP ON TEST
004374  000400                          BR      SAWTH
                                ;THIS TEST IS USED TO GENERATE NOISE
                                ;IN DRIVE ADDRESS CIRCUIT
                                ;THE ROUTINE SEEKS TO CYLINDER N
                                ;THEN TO CYLINDER N+2
```

```
                        ;THEN TO CYLINDER N+1
                        ;THEN TO CYLINDER N+3
                        ;THIS GENERATES A SAW TOOTH AFFECT WHEN
                        ;SEEKING.
                        ;
                        ;
004376  012767  177766  174500  SAWTH:  MOV     #-10.,,PASSC
004404  012706  001000          MOV     #1000,%6        ;SET UP STACK
004410  012777  005022  174416  MOV     #PWAIT,@VECTOR  ;SET UP DISK VECTOR
004416  012777  000340  174412  MOV     #340,@STATUS
004424  016777  174410  174350  MOV     PRIORITY,@CSR   ;SET UP DRIVE #
004432  012767  000100  174430  MOV     #100,SAVE
004440  004567  003516          NOISET: JSR     %5,DSKNOS       ;SET UP CPU STATUS
004444  056777  174440  174350  BIS     WORK,@RKDA      ;SET THE DRIVE NUMBER
004452  042777  017777  174342  BIC     #17777,@RKDA
004460  056777  174404  174334  BIS     SAVE,@RKDA      ;SET FOR FIRST SEEK
004466  012777  000011  174320  MOV     #11,@RKCS       ;GO SEEK
004474  105777  174314          TSTB    @RKCS           ;TEST FOR READY
004500  100375                  BPL     .-4             ;NO FLAG GO BACK
004502  052777  000100  174304  BIS     #BIT6,@RKCS     ;SET INT ENABLE
004510  000777                  BR      .               ;WAIT FOR INTERRUPT FROM SCP
004512  042777  000100  174274  BIC     #BIT6,@RKCS     ;CLEAR INTERRUPT ENABLE
004520  005777  174270          TST     @RKCS           ;TEST FOR ERRORS
004524  100015                  BPL     SAW1            ;NO ERRORS GO ON
004526  017767  174262  174356  MOV     @RKCS,WORK1     ;FETCH RKCS
004534  017767  174262  174346  MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS
004542  012767  000014  174316  ERR14:  MOV     #14,ERCOUNT
004550  004567  006072          JSR     %5,STAER1       ;REPORT ERROR
004554  000167  177616          JMP     SAWTH           ;LOOP ON ERROR
004560  162777  000040  174234  SAW1:   SUB     #40,@RKDA       ;DEC. CYLINDER NO.
004566  012777  000011  174220  MOV     #11,@RKCS       ;GO SEEK
004574  105777  174214          TSTB    @RKCS           ;TEST FOR CONTROL READY
004600  100375                  BPL     .-4             ;NO FLAG GO BACK
004602  052777  000100  174204  BIS     #BIT6,@RKCS     ;SET INT, ENABLE
004610  000777                  BR      .               ;WAIT FOR INT FROM SCP
004612  042777  000100  174174  BIC     #BIT6,@RKCS     ;CLEAR INT ENABLE
004620  005777  174170          TST     @RKCS           ;TEST FOR ERRORS
004624  100015                  BPL     SAW2            ;BRANCH IF NO ERRORS
004626  017767  174162  174256  MOV     @RKCS,WORK1     ;FETCH FUNCTION REGISTER
004634  017767  174162  174246  MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS REGISTER
004642  012767  000015  174216  ERR15:  MOV     #15,ERCOUNT
004650  004567  005772          JSR     %5,STAER1       ;REPORT ERROR
004654  000167  177516          JMP     SAWTH           ;LOOP ON ERROR
004660  062777  000100  174134  SAW2:   ADD     #100,@RKDA      ;SET UP NEXT CYLINDER
004666  012777  000011  174120  MOV     #11,@RKCS       ;GO SEEK
004674  105777  174114          TSTB    @RKCS           ;TEST FOR READY
004700  100375                  BPL     .-4             ;GO BACK NO FLAG
004702  052777  000100  174104  BIS     #BIT6,@RKCS     ;SET INT,ENABLE
004710  000777                  BR      .               ;WAIT FOR INT. FROM SCP
004712  042777  000100  174074  BIC     #BIT6,@RKCS     ;CLR INT. ENABLE
004720  005777  174070          TST     @RKCS           ;TEST FOR ERRORS
004724  100015                  BPL     SAW3            ;BRANCH IF NO ERRORS
004726  017767  174062  174156  MOV     @RKCS,WORK1     ;FETCH FUNCTION REGISTER
004734  017767  174062  174146  MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS
004742  012767  000016  174116  ERR16:  MOV     #16,ERCOUNT
```

```
004750  004567  005672          JSR     X5,STAER1       ;REPORT ERRO
004754  000167  177416          JMP     SAWTH           ;LOOP ON ERROR
004760  062767  000140  174102  SAW3:   ADD     #140,SAVE       ;SET UP NEXT SAW TOOTH
004766  022767  014540  174074          CMP     #14540,SAVE     ;IS IT THE LAST SAW TOOTH
004774  001221                  BNE     NOISET
004776  032777  004000  173774          BIT     #BIT11,@SR      ;CHECK TO LOOP ON TEST
005004  001402                  BEQ     .+6
005006  000167  177364          JMP     SAWTH           ;LOOP ON TEST
005012  005267  174066          INC     PASSC
005016  001205                  BNE     NOISET-6
005020  000413                  BR      ADT3
                                ;
                                ;
                                ;
                                ;
005022  032777  020000  173764  PWAIT:  BIT     #BIT13,@RKCS    ;TEST FOR SEEK COMPLETE
005030  001004                  BNE     XWPAT           ;SEEK NOT COMPLETE
005032  005267  174052          INC     WORK
005036  001401                  BEQ     .+4
005040  000002                  RTI                     ;RETURN TO ADT2X SEEK COMPLETE
005042  062716  000002  XWPAT:  ADD     #2,(6)          ;SEEK NOT COMPLETE
005046  000002                  RTI                     ;RETURN TO WAIT
                                ;
                                ;
```

```
                        ;
                        ;THIS IS A LOOK AHEAD TEST ,
                        ;IT FIRST LOOKS FOR ADDRESS ZERO BY READING THE RKDS REGISTER
                        ;THEN CONTINUING TO READ THE RKDS REGISTER IT LOOKS FOR AN ADRESS
                        ;WHICH IS EQUAL TO OR ONE GREATER THAN THE PREVIOUS.
                        ;
                        ;
                        ;;
005050  012706  001000          ADT3:   MOV     #1000,%6        ;SET UP STACK
005054  012777  000001  173732          MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
005062  105777  173726          TSTB    @RKCS
005066  100375                  BPL     .-4
005070  004567  003066          JSR     %5,DSKNOS       ;SET THE DISK NUMBER
005074  056777  174010  173720          BIS     WORK,@RKDA      ;SET THE DRIVE NUMBER
005102  005067  173762          CLR     SAVE
005106  005067  174000          CLR     WORK1
005112  032777  000400  173706  TSTZER: BIT     #BIT8,@RKDS     ;CHECK DRIVE STATUS OK
005120  001774                  BEQ     .-6
005122  032777  000017  173676          BIT     #17,@RKDS       ;TEST FOR SECTOR 0
005130  001416                  BEQ     TSTNXS
005132  005267  173754          INC     WORK1           ;WAIT FOR SECTOR ZERO
005136  005767  173750          TST     WORK1
005142  001363                  BNE     TSTZER
005144  012767  000017  173714          MOV     #17,ERCOUNT     ;COULD NOT FIND SECTOR 0
005152  017767  173650  173730          MOV     @RKDS,WORK
005160  004567  005422          ERR17:  JSR     %5,STAER        ;REPORT ERROR
005164  000731                  BR      ADT3            ;LOOP ON ERROR
005166  032777  000400  173632  TSTNXS: BIT     #BIT8,@RKDS     ;TST FOR DRIVE STATUS OK
005174  001774                  BEQ     .-6
005176  017767  173624  173704          MOV     @RKDS,WORK
005204  042767  177760  173676          BIC     #177760,WORK
005212  016767  173672  173672          MOV     WORK,WORK1
005220  026767  173644  173662          CMP     SAVE,WORK
005226  001757                  BEQ     TSTNXS          ;IT WAS THE SAME ADDRESS
005230  166767  173634  173652          SUB     SAVE,WORK
005236  022767  000001  173644          CMP     #1,WORK         ;WAS THE ADDR, ONE GREATER
005244  001007                  BNE     LKADER          ;ADDR GREATER THAN +1
005246  022767  000012  173614          CMP     #12,SAVE        ;IS IT THE LAST SECTOR
005254  001414                  BEQ     LPAD3           ;YES EXIT
005256  005267  173606          INC     SAVE
005262  000741                  BR      TSTNXS          ;CHECK NEXT SECTOR
005264  012767  000020  173574  LKADER: MOV     #20,ERCOUNT
005272  016767  173572  173610          MOV     SAVE,WORK
005300  004567  005342          ERR20:  JSR     %5,STAER1
005304  000661                  BR      ADT3
005306  032777  004000  173464  LPAD3:  BIT     #BIT11,@SR      ;CHECK TO LOOP ON TEST
005314  001255                  BNE     ADT3            ;YES LOOP ON TEST
```

```
005316  005067  173530          DATATX: CLR     DMA
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
005322  016767  173536  173516  DATAT:  MOV     SWRDCT,WRDCT
005330  012777  006636  173476          MOV     #DKINT,@VECTOR
005336  012777  000340  173472          MOV     #340,@STATUS
005344  012767  005364  173526          MOV     #LDAT,HRDER     ;SETUP FOR HARD ERROR
005352  012777  000340  173422          MOV     #340,@CSR       ;LOCK UP PROCESSOR PRIORITY
005360  004567  003012                  JSR     %5,PASEL        ;SET UP DATA BUFFERS
005364  004567  002010          LDAT:   JSR     %5,OPDSEL       ;SET UP DISK ADDRESS
005370  032767  040000  173444          BIT     #BIT14,FLAG     ;TEST FOR WRITE
005376  001421                          BEQ     ESH             ;TEST FOR READ
005400  012767  014536  173450          MOV     #OUTBUF,BUF     ;SETUP OUTPUT BUFFER
005406  104503                          WRITE   +100            ;WRITE WITH INT. ENABLED
005410  032777  001000  173362          BIT     #BIT9,@SR       ;FIND OUT HOW TO WAIT FOR INT
005416  001002                          BNE     WRWAIT          ;WAIT WITH WAIT INSTRUCTION
005420  000004                          IOT                     ;WAIT IN BACKGROUND TEST
005422  000404                          BR      XSLH
005424  016777  173410  173350  WRWAIT: MOV     PRIORITY,@CSR
005432  000001                          WAIT                    ;WAIT FOR FLAG
005434  004767  002024          XSLH:   JSR     %7,DISBUF       ;SET BUFFER FOR WRITE
005440  000751                          BR      LDAT
005442  004567  001732          ESH:    JSR     %5,OPDSEL       ;OPERATOR SELECTED CYLINDER?
005446  032767  010000  173366          BIT     #BIT12,FLAG     ;TEST FOR READ
005454  001451                          BEQ     RDCKLP
005456  016767  173424  173372          MOV     INBUF,BUF       ;SETUP OUTPUT BUFFER
005464  042767  000003  173350          BIC     #3,FLAG         ;CLEAR RE-READ COUNT
```

```
005472  012777  000340  173302  DSKRD:  MOV     #340,@CSR
005500  005267  173336                  INC     FLAG
005504  104505                          READ    +100            ;READ + INT ENABLE
005506  032777  001000  173264          BIT     #BIT9,@SR       ;FIND OUT HOW TO WAIT FOR INT.
005514  001002                          BNE     RDWAIT          ;WAIT WITH WAIT INSTRUCTION
005516  000004                          IOT
005520  000404                          BR      ELH
005522  016777  173312  173252  RDWAIT: MOV     PRIORITY,@CSR   ;SET UP PRIORITY
005530  000001                          WAIT                    ;WAIT FOR FLAG
005532  032777  010000  173240  ELH:    BIT     #BIT12,@SR
005540  001002                          BNE     ADRD
005542  004567  003064                  JSR     %5,COMPARE      ;COMPARE OUTBUFFER TO INBUFFER
005546  016767  173270  173334  ADRD:   MOV     FLAG,WORK       ;CHECK DISK RE-READ COUNT
005554  042767  177774  173326          BIC     #177774,WORK    ;DO 3 RE-READS.
005562  022767  000003  173320          CMP     #3,WORK
005570  001340                          BNE     DSKRD           ;DO ANOTHER RE-READ
005572  004767  001666                  JSR     %7,DISBUF       ;GO SET UP DISK BUFFER.
005576  000721                          BR      ESH
005600  004567  001574  RDCKLP: JSR     %5,OPDSEL
005604  032767  000100  173230          BIT     #BIT6,FLAG
005612  001421                          BEQ     SLH
005614  012777  000340  173160          MOV     #340,@CSR
005622  104513                          RDCK    +100
005624  032777  001000  173146          BIT     #BIT9,@SR
005632  001002                          BNE     RCHWAIT
005634  000004                          IOT
005636  000404                          BR      ESRDCH
005640  016777  173174  173134  RCHWAIT: MOV    PRIORITY,@CSR
005646  000001                          WAIT
005650  004767  001610          ESRDCH: JSR     %7,DISBUF
005654  000751                          BR      RDCKLP
005656  004567  001516          SLH:  JSR       %5,OPDSEL       ;IS THE OPERATOR SELECTING THE TRACK
005662  032767  020000  173152          BIT     #BIT13,FLAG     ;TEST FOR WRITE CHECK
005670  001424                          BEQ     MSTR
005672  012767  014536  173156          MOV     #OUTBUF,BUF     ;SET UP CURRENT ADDRESS
005700  012777  000340  173074          MOV     #340,@CSR
005706  104507                  WRCHECK +100
005710  032777  001000  173062          BIT     #BIT9,@SR
005716  001002                          BNE     WCWAIT          ;WAIT FOR FLAG IN WAIT INST.
005720  000004                          IOT                     ;WAIT FOR FLAG IN BACKGROUND TEST
005722  000404                          BR      XESH
005724  016777  173110  173050  WCWAIT: MOV     PRIORITY,@CSR
005732  000001                          WAIT                    ;WAIT FOR THE FLAG
005734  004767  001524          XESH:  JSR      %7,DISBUF       ;SET UP THE DISK BUFFER
005740  000746                          BR      SLH
005742  032777  000400  173030  MSTR:   BIT     #BIT8,@SR       ;ARE WE LOOPING ON AN ADDRESS
005750  001402                          BEQ     .+6             ;NO!ADDRESS SELECTION UNDER PROGRAM CONT
005752  000167  177344                  JMP     DATAT           ;LOOP ON ADDRESS
005756  005767  173060                  TST     FLAG
005762  100002                          BPL     .+6             ;UNDER PROGRAM CONTROL
005764  000167  000520                  JMP     EXTPP           ;OPERATOR SELECTED PATTERN
005770  062767  000002  173056          ADD     #2,PATNU        ;INC PATTERN INDEX
005776  022767  000044  173050          CMP     #44,PATNU
006004  001402                          BEQ     .+6
```

```
006006  000167  177310                  JMP     DATAT           ;NOT LAST PATTERN EXIT
006012  005067  173036                  CLR     PATNU           ;LAST PATTERN EXIT

                                ;THIS IS A RANDOM ADDRESS RANDOM DATA TEST
                                ;
                                ;
                                ;
006016  012767  177000  173060  RANEX:  MOV     #-1000,PASSC    ;SET UP PASS COUNT
006024  042767  001003  173010          BIC     #1003,FLAG      ;CLEAR ERROR FLAG BITS
006032  012706  001000                  MOV     #1000,%6        ;SET UP STACK
006036  012767  006330  173034          MOV     #RANER,HRDER    ;SET UP FOR HARD ERROR
006044  012777  006636  172762          MOV     #DKINT,@VECTOR  ;SET UP INTERRUPT VECTOR
006052  012777  000340  172756          MOV     #340,@STATUS
006060  016777  172754  172714          MOV     PRIORITY,@CSR   ;SET PRIORITY TO LEVEL 5
006066  012767  000001  173014  WRLG:   MOV     #1,WORK         ;SET UP RANDOM GENERATOR WORD
006074  012701  014536                  MOV     #OUTBUF,%1
006100  004567  002332                  JSR     %5,RANDOM       ;GENERATE RANDOM DATA
006104  016767  006426  172740          MOV     OUTBUF,DMA      ;SET UP DISK ADDRESS
006112  042767  160000  172732          BIC     #160000,DMA
006120  022767  014533  172724          CMP     #14533,DMA
006126  101757                          BLOS    WRLG
006130  042767  177760  006400          BIC     #177760,OUTBUF
006136  022767  000014  006372          CMP     #14,OUTBUF      ;IS IT THE LAST SECTOR
006144  101750                          BLOS    WRLG            ;BRALCH IF YES
006146  012767  000200  172672          MOV     #200,WRDCT      ;SET UP WOPD COUNT =1SECTOP
006154  016767  172666  172726          MOV     WRDCT,WORK      ;GENERATE RANDOM BUFFER
006162  012701  014536                  MOV     #OUTBUF,%1
006166  004567  002244                  JSR     %5,RANDOM
006172  012777  000001  172614          MOV     #BIT0,@RKCS     ;CLEARTHE CONTROL
006200  105777  172610                  TSTB    @RKCS           ;IS THE CONTROL READY
006204  100375                          BPL     .-4
006206  012767  014536  172642          MOV     #OUTBUF,BUF     ;SET UP OUTPUT BUFFER
006214  104503                          WRITE   +100    ;
006216  032777  001000  172554          BIT     #BIT9,@SR
006224  001002                          BNE     .+6
006226  000004                          IOT
006230  000404                          BR      .+12
006232  016777  172602  172542          MOV     PRIORITY,@CSR   ;LOCK UP CPU
006240  000001                          WAIT
006242  104507                          WRCHECK +100            ;WRITE CHECK THE DISK
006244  032777  001000  172526          BIT     #BIT9,@SR
006252  001002                          BNE     .+6
006254  000004                          IOT
006256  000404                          BR      .+12
006260  016777  172554  172514          MOV     PRIORITY,@CSR
006266  000001                          WAIT

006270  016767  172612  172560          MOV     INBUF,BUF
006276  104505                          READ    +100
006300  032777  001000  172472          BIT     #BIT9,@SR
006306  001002                          BNE     .+6
006310  000004                          IOT
006312  000404                          BR      .+12
006314  016777  172520  172460          MOV     PRIORITY,@CSR
006322  000001                          WAIT
```

```
006324  004567  002302                  JSR     %5,COMPARE      ;COMPARE THE DATA FOR ERRORS
006330  032767  001000  172504  RANER:  BIT     #BIT9,FLAG      ;TEST FOR ERRORS
006336  001422                          BEQ     EXRAX           ;NO ERRORS
006340  042767  001000  172474          BIC     #BIT9,FLAG
006346  004567  004546                  JSR     %5,CONV         ;SET UP FOR ADDRESS REPORT
006352  001052                          DMA
006354  013541                          MES1
006356  000006                          6
006360  004567  004534                  JSR     %5,CONV         ;SET UP WOPD COUNT FOP PEPORT
006364  001046                          WRDCT
006366  013625                          MES3
006370  000006                          6
006372  104000                          EMT+0
006374  013523                          HED6
006376  013541                          MES1
006400  013625                          MES3
006402  177777                          -1
006404  005267  172474          EXRAX:  INC     PASSC           ;+1 PASS COUNT
006410  001226                          BNE     WRLG            ;BRANCH IF TEST NNT OVER
006412  032777  004000  172360          BIT     #BIT11,@SR      ;LOOP ON TEST
006420  001402                          BEQ     .+6
006422  000167  177370                  JMP     RANEX           ;YES LOOP ON TEST
                                ;
                                ;CHECK FOR MULTI DISK MODE
                                ;IF IN MULTI DISK MODE REPORT "END"
                                ;IF LAST DISK ON SYSTEM HAS BEEN
                                ;EXERCISED.
                                ;
                                ;
006426  005067  172420                  CLR     DMA
006432  032767  004000  172402          BIT     #BIT11,FLAG     ;ARE WE IN MULTI DISK MODE
006440  001416                          BEQ     REPOEN          ;REPORT "END"
006442  016767  172374  172440          MOV     FLAG,WORK       ;WHAT DISK ARE WE ON
006450  042767  177743  172432          BIC     #177743,WORK    ;IF LAST DISK ON SYSTEM
006456  026767  172426  172412          CMP     WORK,DSKNOR     ;REPORT END
006464  001404                          BEQ     REPOEN          ;REPORT "END" LAST DISK
006466  062767  000004  172346          ADD     #4,FLAG         ;INC. DISK NO.
006474  000405                          BR      EXTPP           ;EXERCISE DISK
006476  104001                  REPOEN: EMT     -1
006500  014530                          END                     ;REPORT END OF PASS
006502  042767  000034  172332          BIC     #34,FLAG        ;RESET DRIVE NUMBER TO ZERO
006510  000167  173376          EXTPP:  JMP     ADTST           ;RECYCLE
                                ;
                                ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
                                ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
                                ;BYTE OF THE CONTROL REGISTER.
                                ;
006514  016705  172302          DISK:   MOV     RKDA,%5         ;SET UP TO LOAD DISK REG
006520  016715  172326                  MOV     DMA,(5)         ;LOAD WORD ADDRESS
006524  016767  172312  172356          MOV     FLAG,WORK
006532  000241                          CLC                     ;SET UP THE DRIVE NUMBER
006534  006067  172350                  ROR     WORK
006540  006067  172344                  ROR     WORK
006544  006067  172340                  ROR     WORK
006550  006067  172334                  ROR     WORK
```

```
006554  006067  172330                  ROR     WORK
006560  006067  172324                  ROR     WORK
006564  042767  017777  172316          BIC     #17777,WORK
006572  056715  172312                  BIS     WORK,(5)
006576  016745  172254                  MOV     BUF,-(5)        ;SET UP CURRENT ADDRESS
006602  016745  172240                  MOV     WRDCT,-(5)      ;LOAD WORD COUNT
006606  005415                          NEG     (5)
006610  011604                          MOV     (6),%4
006612  014467  172272                  MOV     -(4),WORK
006616  042767  177600  172264          BIC     #177600,WORK
006624  016745  172260                  MOV     WORK,-(5)       ;LOAD FUNCTION REG.
006630  052715  000400                  BIS     #BIT8,(5)       ;SET STOP ON SOFT ERROR
006634  000002                          RTI                     ;RETURN FROM TRAP
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;RK11 DISK INTERRUPT HANDLER
                                ;ROUTINE CONTINUES ON ERRORS
                                ;
006636  042777  000020  172136  DKINT:  BIC     #BIT4,@CSR      ;CLEAR TRACE BIT
006644  005777  172144                  TST     @RKCS           ;TEST FOR ERROR
006650  100402                          BMI     .+6
006652  000167  000472                  JMP     INTEXT          ;JUMP IF NO ERRORS
006656  052767  001000  172156          BIS     #BIT9,FLAG      ;SET ERROR BIT
006664  017767  172134  172216          MOV     @RKER,WORK      ;REPORT ERROR
006672  004567  004222                  JSR     %5,CONV         ;CONVERT TO ASCII
006676  001110                          WORK
006700  013556                          MES1A
006702  000006                          6
006704  104001                          EMT+1
006706  013433                          HED2
006710  032777  000002  172076          BIT     #BIT1,@RKCS     ;CHECK FOR READ
006716  001015                          BNE     DELMES          ;IF READING REPORT WHICH READ
006720  016767  172116  172162          MOV     FLAG,WORK
006726  042767  177774  172154          BIC     #177774,WORK
006734  004567  004160                  JSR     %5,CONV
006740  001110                          WORK
006742  013746                          MES13
006744  000001                          1
006746  104001                          EMT+1
006750  013746                          MES13
```

```
006752  005777  172046          DELMES: TST     @RKER           ;TEST FOR DRIVE ERROR
006756  100012                          BPL     REDA            ;BRANCH NO DRIVE ERROR
006760  017767  172042  172122        MOV     @RKDS,WORK      ;FETCH DRIVE STATUS
006766  004567  004126                JSR     %5,CONV
006772  001110                        WORK
006774  013610                        MES2A
006776  000006                        6
007000  104001                        EMT+1
007002  013610                        MES2A
007004  017767  172012  172076  REDA:  MOV     @RKDA,WORK      ;SET UP 16 BITS OF ADDR.
                                ;ALTER ADDRESS FOR CORRECT ERROR REPORT
007012  032777  040000  171774        BIT     #BIT14,@RKCS    ;WAS IT A DRIVE ERROR
007020  001036                        BNE     ADRREP          ;YES DO NOT ALTER ADDRESS
007022  016767  172062  172062        MOV     WORK,WORK1
007030  042767  177760  172054        BIC     #177760,WORK1   ;MASK SECTOR
007036  032767  000017  172046        BIT     #17,WORK1       ;TEST FOR ERROR IN LAST SECTOR
007044  001403                        BEQ     .+10            ;ERROR IN LAST SECTOR
007046  005367  172036                DEC     WORK            ;ERROR NOT IN LAST SECTOR
007052  000421                        BR      ADRREP          ;REPORT ERROR
007054  052767  000013  172026        BIS     #13,WORK        ;SET UP FOR ERROR IN LAST SECTOR
007062  032767  000020  172020        BIT     #BIT4,WORK      ;TEST FOR WHICH SURFACE
007070  001404                        BEQ     .+12            ;ERROR ON SURFACE ONE
007072  042767  000020  172010        BIC     #BIT4,WORK      ;ERROR ON SURFACE ZERO
007100  000406                        BR      ADRREP          ;REPORT ERROR
007102  052767  000020  172000        BIS     #BIT4,WORK      ;ERROR ON SURFACE ONE
007110  162767  000040  171772        SUB     #40,WORK        ;SUB. FROM CYLINDER NUMBER
007116  004567  003776          ADRREP: JSR     %5,CONV         ;CONVERT TO ASCII
007122  001110                        WORK
007124  013541                        MES1
007126  000006                        6
007130  104000                        EMT+0
007132  013556                        MES1A
007134  013541                        MES1
007136  177777                        -1
007140  017767  171650  171742        MOV     @RKCS,WORK      ;SET UP STATUS
007146  004567  003746                JSR     %5,CONV
007152  001110                        WORK
007154  013573                        MES2
007156  000006                        6
007160  104001                        EMT+1
007162  013573                        MES2
007164  032777  002000  171634        BIT     #BIT10,@RKDS    ;TEST FOR DRIVE UNSAFE
007172  001402                        BEQ     .+6             ;DRIVE SAFE CONTINUE
007174  000000                        HALT                    ;DRIVE UNSAFE HALT
007176  000777                        BR      .               ;LOCK UP TEST
007200  032777  040000  171606  CKHRDER:BIT    #BIT14,@RKCS    ;TEST FOR HARD ERROR
007206  001436                        BEQ     SOFTER          ;GO AND CONTINUE SOFT ERROR
007210  012706  001000          HRDEXIT:MOV   #1000,%6
007214  032777  002000  171556        BIT     #BIT10,@SR      ;HALT ON ERROR
007222  001401                        BEQ     .+4
007224  000000                        HALT                    ;YES HALT BIT 10 SET IN SR
007226  012777  000001  171560        MOV     #BIT0,@RKCS
007234  105777  171554                TSTB    @RKCS
007240  100375                        BPL     .-4
```

```
007242  004567  000714           JSR     %5,DSKNOS       ;SET UP THE DRIVE NUMBER
007246  012777  000015  171540   MOV     #15,@RKCS       ;RESTORE DRIVE
007254  032777  000100  171544   BIT     #BIT6,@RKDS     ;IS THE SEARCH COMPLETE
007262  001774                   BEQ     .-6
007264  012777  000001  171522   MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
007272  105777  171516           TSTB    @RKCS           ;IS THE CONTROL READY
007276  100375                   BPL     .-4
007300  000177  171574           JMP     @HRDER          ;EXIT HARD ERROR
007304  005777  171506  SOFTER:  TST     @RKWC           ;CHECK FOR X-FER DONE
007310  001417                   BEQ     INTEXT          ;EXIT FROM ROUTINE
007312  032777  001000  171460   BIT     #BIT9,@SR
007320  001402                   BEQ     .+6
007322  162716  000002           SUB     #2,(6)          ;X-FER NOT DONE SET UP FOR RETURN
007326  032777  002000  171444   BIT     #BIT10,@SR      ;HALT ON ERROR
007334  001401                   BEQ     .+4
007336  000000                   HALT                    ;YES HALT BIT10 SET IN SR
007340  052777  000001  171446   BIS     #BIT0,@RKCS     ;SET GO AND CONTINUE
007346  000002                   RTI                     ;RETURN TO WAIT INSTR.
                         ;
                         ;
007350  032777  020000  171422  INTEXT:  BIT     #BIT13,@SR      ;HALT ON COMPLETION FLAG
007356  001401                   BEQ     .+4
007360  000000                   HALT                    ;YES BIT 13 SET IN SR HALT
007362  032777  001000  171410   BIT     #BIT9,@SR
007370  001002                   BNE     .+6
007372  012706  000774           MOV     #774,%6         ;RESET STACK
007376  000002                   RTI                     ;EXIT
```

```
                                ;ROUTINE TO SET UP CYLINDER # FROM SR
                                ;CYLINDER NO. IN SR 7-0
                                ;ENTER FROM JSR %5, OPDSEL
                                ;
007400  032767  000040  171434  OPDSEL: BIT     #BIT5,FLAG      ;DOES SR CONTAIN CYLINDER #
007406  001001                        BNE     .+4
007410  000205                        RTS     %5              ;NO! EXIT
007412  017767  171362  171470        MOV     @SR,WORK        ;FETCH SR
007420  042767  177400  171462        BIC     #177400,WORK    ;MASK CYLINDER BITS
007426  000241                        CLC
007430  006167  171454                ROL     WORK
007434  006167  171450                ROL     WORK
007440  006167  171444                ROL     WORK
007444  006167  171440                ROL     WORK
007450  006167  171434                ROL     WORK
007454  056767  171430  171370        BIS     WORK,DMA        ;LOAD CYLINDER BITS
007462  000205                        RTS     %5              ;EXIT
```

```
                                ;
                                ;ROUTINE TO SETUP DISK BUFFERS
                                ;ADD WORD COUNT TO STARTING DISK ADDRESSES
                                ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
                                ;
007464  032777  000400  171306 DISBUF: BIT     #BIT8,@SR
007472  001401                       BEQ     .+4
007474  000506                       BR      BUFEXIT+4       ;LOOPING ON A SELECTED ADDRESS
007476  004767  000530               JSR     %7,BLSZ         ;DEFINE BLOCK SIZE
007502  016767  171374  171402       MOV     BLOCK,WORK1
007510  016767  171336  171372 INCSEC: MOV     DMA,WORK
007516  042767  177760  171364       BIC     #177760,WORK    ;MASK OUT SECTOR COUNT
007524  022767  000013  171356       CMP     #13,WORK        ;CHECK FOR LAST SECTOR
007532  001406                       BEQ     INCSUR          ;CHECK SURFACE
007534  005267  171312               INC     DMA             ;+1 SECTOR COUNT
007540  005367  171336       DECBLK: DEC     BLOCK           ;-1 FROM BLOCK COUNT
007544  001423                       BEQ     COMDAR          ;CMP DMA TO RKDA
007546  000760                       BR      INCSEC          ;RECYCLE
007550  042767  000017  171274 INCSUR: BIC     #17,DMA         ;FETCH ADDRESS
007556  032767  000020  171266       BIT     #BIT4,DMA       ;CHECK FOR SURFACE
007564  001407                       BEQ     SWSUR           ;+1 SURFACE
007566  062767  000040  171256       ADD     #40,DMA         ;INC. CYLINDER NUMBER
007574  042767  000020  171250       BIC     #BIT4,DMA       ;CLEAR SURFACE
007602  000756                       BR      DECBLK          ;RECYCLE
007604  052767  000020  171240 SWSUR: BIS     #BIT4,DMA
007612  000752                       BR      DECBLK
007614  004567  000342       COMDAR: JSR     %5,DSKNOS       ;SET UP THE DRIVE NUMBER
007620  056767  171264  171224       BIS     WORK,DMA
007626  026777  171220  171166       CMP     DMA,@RKDA       ;COMPARE LOWER 16 BITS
007634  001421                       BEQ     CMDAE
007636  004567  003256       ERADR:  JSR     %5,CONV         ;CONVERT DMA REG COUNT TO ASCII
007642  001052                       DMA
007644  013541                       MES1
007646  000006                       6
007650  104000                       EMT     +0              ;REPORT ERROR
007652  013452                       HED4
007654  013541                       MES1
007656  177777                       -1
007660  005067  171166               CLR     DMA             ;DISK ADDRESS ERROR RE-START PROG.
007664  032777  002000  171106       BIT     #BIT10,@SR      ;HALT ON ERROR
007672  001401                       BEQ     .+4
007674  000000                       HALT                    ;SWITCH 10 SET IN SR HALT
007676  000207                       RTS     %7
                                ;
007700  105767  171136       CMDAE:  TSTB    FLAG            ;CHECK FOR LAST DISK BUFFER
007704  100013                       BPL     BUFINX
007706  005067  171140       BUFEXIT: CLR    DMA             ;CLEAR ADDRESS BITS
007712  062716  000002               ADD     #2,(6)          ;INC STACK POINTER
007716  042767  000200  171116 AKH:  BIC     #200,FLAG
007724  016767  171134  171114       MOV     SWRDCT,WRDCT
007732  000512                       BR      EXTDR           ;EXIT
007734  016767  171112  171106 BUFINX: MOV     DMA,TRACK       ;FETCH CYLINDER NUMBER
007742  042767  160037  171100       BIC     #160037,TRACK   ;MASK CYLINDER NUMBER
007750  005067  171140               CLR     WORK2
```

```
007754  016767  171072  171126          MOV     DMA,WORK
007762  042767  177760  171120  XINCSEC: BIC     #177760,WORK    ;MASK SECTOR BITS
007770  005267  171120                  INC     WORK2
007774  022767  000013  171106          CMP     #13,WORK        ;CMP FOR LAST SECTOR
010002  001406                          BEQ     XINCSUR         ;+1 SURFACE
010004  005267  171100                  INC     WORK
010010  005367  171076  CONTINC: DEC     WORK1
010014  001362                          BNE     XINCSEC
010016  000737                          BR      AKH
010020  042767  000017  171062  XINCSUR: BIC     #17,WORK
010026  032767  000020  171016          BIT     #BIT4,DMA       ;TEST SURFACE
010034  001004                          BNE     CLRSUR
010036  052767  000020  171044          BIS     #BIT4,WORK
010044  000761                          BR      CONTINC
010046  042767  000020  171034  CLRSUR: BIC     #BIT4,WORK      ;CLEAR SURFACE BIT
010054  062767  000040  170766          ADD     #40,TRACK       ;SET UP TEMP CYLINDER NUMBER
010062  022767  014500  170760          CMP     #14500,TRACK    ;ARE WE AT THE LAST CYLINDER
010070  001347                          BNE     CONTINC         ;BRANCH IF NO!
010072  016767  171016  170746          MOV     WORK2,WRDCT
010100  000241                          CLC
010102  006167  170740                  ROL     WRDCT
010106  006167  170734                  ROL     WRDCT
010112  006167  170730                  ROL     WRDCT
010116  006167  170724                  ROL     WRDCT
010122  006167  170720                  ROL     WRDCT
010126  006167  170714                  ROL     WRDCT
010132  006167  170710                  ROL     WRDCT
010136  032777  004000  170662          BIT     #BIT11,@RKDS    ;TEST FOR DENSITY
010144  001402                          BEQ     .+6
010146  006167  170674                  ROL     WRDCT
010152  052767  000200  170662          BIS     #200,FLAG
010160  000207          EXTDR:  RTS     %7              ;EXIT
                        ;ROUTINE TO SET UP THE DISK NUMBER FOR THE ADDRESS TEST
                        ;
                        ;
010162  016767  170654  170720  DSKNOS: MOV     FLAG,WORK       ;FETCH THE FLAG WORD
010170  006067  170714                  ROR     WORK
010174  006067  170710                  ROR     WORK
010200  000241                          CLC
010202  006067  170702                  ROR     WORK
010206  006067  170676                  ROR     WORK
010212  006067  170672                  ROR     WORK
010216  006067  170666                  ROR     WORK
010222  042767  017777  170660          BIC     #17777,WORK     ;MASK THE DISK NUMBER
010230  000205                          RTS     %5              ;EXIT
                        ;
                        ;
                        ;
```

```
                                ;
                                ;
                                ;
                                ;
                                ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
                                ;
010232  032777  004000  170566  BLSZ:   BIT     #BIT11,@RKDS    ;CHECK DENSITY
010240  001404                          BEQ     LODEN           ;DRIVE IS LOW DENSITY
010242  012767  000377  170632          MOV     #377,BLOCK      ;DRIVE IS HI DENSITY
010250  000403                          BR      SETBLK          ;SET UP NO. OF WORDS PER BLOCK
010252  012767  000177  170622  LODEN:  MOV     #177,BLOCK      ;SET UP NO. WORDS PER BLOCK
010260  016767  170562  170622  SETBLK: MOV     WRDCT,WORK      ;FETCH WORD COUNT
010266  036767  170610  170614          BIT     BLOCK,WORK
010274  001410                          BEQ     RORBLK
010276  046767  170600  170604          BIC     BLOCK,WORK      ;SET UP BLOCK OVERFLOW
010304  005267  170572                  INC     BLOCK
010310  066767  170566  170572          ADD     BLOCK,WORK
010316  006067  170566  RORBLK: ROR     WORK
010322  006067  170562                  ROR     WORK
010326  006067  170556                  ROR     WORK
010332  006067  170552                  ROR     WORK
010336  006067  170546                  ROR     WORK
010342  006067  170542                  ROR     WORK
010346  006067  170536                  ROR     WORK
010352  032777  004000  170446          BIT     #BIT11,@RKDS    ;TEST FOR HI DENSITY
010360  001402                          BEQ     .+6             ;LOW DENSITY
010362  006067  170522                  ROR     WORK
010366  016767  170516  170506          MOV     WORK,BLOCK      ;BLOCK COUNT
010374  000207                          RTS     %7              ;EXIT
                                ;
                                ;
                                ;
                                ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
                                ;
                                ;ENTER FROM JSR %5 PASEL
                                ;
010376  016700  170452          PASEL:  MOV     PATNU,%0        ;SET UP PATTERN NUMBER
010402  016767  170440  170500          MOV     WRDCT,WORK      ;SET UP WORK
010410  012701  014536                  MOV     #OUTBUF,%1      ;LOC. OF OUTBUFFER
010414  022700  000042                  CMP     #42,%0          ;TEST FOR RANDOM DATA NUMBER
010420  001406                          BEQ     RANDOM          ;GO GENERATE RANDOM DATA
010422  016021  010570  FILDAT: MOV     PAT0(0),(1)+    ;FILL BUFFER
010426  005367  170456                  DEC     WORK            ;DEC. WORK COUNT
010432  001373                          BNE     FILDAT          ;LOAD NEXT WORD
010434  000205                          RTS     %5              ;BUFFER FULL
                                ;
                                ;RANDOM DATA
                                ;
                                ;
                                ;
                                ;RANDOM DATA GENERATOR SUBROUTINE
010436  016700  000122  RANDOM: MOV     LONUM,%0        ;SET UP R0 WITH 5 DIGITS LOW
010442  016704  000120          MOV     HINUM,%4        ;SET UP R1 WITH 5 DIGITS HIGH
```

```
010446  012703  000007          MOV     #7,%3           ;SET UP SHIFT COUNT
010452  005002                  CLR     %2              ;CLEAR R2
010454  006300          SHIFT:  ASL     %0              ;SHIFT R0 LEFT AND
010456  006104                  ROL     %4              ;ROTATE CARRY INTO LSB OF R1 INTO
010460  006102                  ROL     %2              ;ROTATE CARRY OUT OF R1 INTO R2
010462  005303                  DEC     %3              ;DECREMENT R3
010464  001373                  BNE     SHIFT           ;CONTINUE SHIFT LOOP
010466  066700  000072          ADD     LONUM,%0        ;ADDN IN NUMBER TO MAKE X 129
010472  005504                  ADC     %4              ;PROPOGATE CARRY
010474  066704  000066          ADD     HINUM,%4        ;ADDN IN NUMBER TO MAKE X 129
010500  005502                  ADC     %2              ;PROPOGATE CARRY
010502  062700  001057          ADD     #1057,%0        ;ADDN LOW CONSTANT
010506  005504                  ADC     %4              ;PROPOGATE CARRIES
010510  005502                  ADC     %2              ;PROPOGATE AGAIN
010512  062704  047401          ADD     #47401,%4       ;ADDN HIGH CONSTANT
010516  005502                  ADC     %2              ;PROPOGATE CARRY
010520  062702  000006          ADD     #6,%2           ;ADDN HIGHEST CONSTANT
010524  062700  000002          ADD     #2,%0           ;REPRIME R0 WITH HIGH DIGIT
010530  005504                  ADC     %4              ;PROPOGATE CARRY
010532  010067  000026          MOV     %0,LONUM        ;PUT R0 BACK IN LONUM
010536  010021                  MOV     %0,(1)+         ;HOLD LONUM FOR PROGRAM
010540  005367  170344          DEC     WORK
010544  001406                  BEQ     EXGEN
010546  010467  000014          MOV     %4,HINUM        ;PUT R1 BACK IN HINUM
010552  010421                  MOV     %4,(1)+         ;HOLD HINUM FOR PROGRAM
010554  005367  170330          DEC     WORK
010560  001326                  BNE     RANDOM
010562  000205          EXGEN:  RTS     %5              ;RETURN TO PROGRAM
010564  000000          LONUM:  0
010566  000000          HINUM:  0
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
```

```
        010570                          .EVEN
                                        ;
                                        ;RK11 DATA PATTERNS
                                        ;
010570  000000                  PAT0:   0
010572  177777                  PAT1:   177777
010574  134510                  PAT2:   134510
010576  043267                  PAT3:   043267
010600  100000                  PAT4:   100000
010602  107070                  PAT5:   107070
010604  070707                  PAT6:   070707
010606  052525                  PAT7:   052525
010610  125252                  PAT10:  125252
010612  177737                  PAT11:  177737
010614  004102                  PAT12:  004102
010616  136363                  PAT13:  136363
010620  063636                  PAT14:  063636
010622  000001                  PAT15:  000001
010624  100005                  PAT16:  100005
010626  000520                  PAT17:  000520
010630  030303                  PAT20:  030303
                                ;PAT21  RANDOM DATA
                                ;
                                ;
                                ;
                                ;DATA COMPARISON ROUTINE
                                ;IF AN ERROR OCCURS BETWEEN THE OUT-BUFFER AND
                                ;THE IN-BUFFER AN ERROR WILL BE REPORTED IN THE
                                ;
                                ;
010632  012767  177766  170226  COMPARE:MOV     #-10,,ERCOUNT   ;REPORT ONLY 10 ERRORS PER BUFFER
010640  005067  170250          CLR     WORK2           ;WORD COUNT
010644  005067  170242          CLR     WORK1
010650  016767  170176  170232  MOV     DMA,WORK
010656  032777  004000  170142  BIT     #BIT11,@RKDS
010664  001404                  BEQ     BLNK
010666  012767  000400  170206  MOV     #400,BLOCK      ;HI DENSITY DRIVE
010674  000403                  BR      .+10
010676  012767  000200  170176  BLNK:   MOV     #200,BLOCK      ;LO DENSITY DRIVE
010704  012767  014536  170156  MOV     #OUTBUF,SAVE    ;SET UP OUTBUFFER POINTER
010712  016767  170170  170152  MOV     INBUF,SAV1      ;SET UP IN BUFFER POINTER
010720  027777  170144  170144  WRDCMP: CMP     @SAVE,@SAV1     ;COMPARE BUFFERS
010726  001051                  BNE     WDERR           ;WORD IN ERROR
010730  005267  170160          WRDINC: INC     WORK2           ;+1 WORD COUNT
010734  005267  170152          INC     WORK1
010740  026767  170102  170146  CMP     WRDCT,WORK2     ;IS COMPLETE BUFFER CHECKED
010746  001440                  BEQ     ADAM            ;EXIT ROUTINE
010750  036767  170126  170134  BIT     BLOCK,WORK1
010756  001425                  BEQ     BLAD1
010760  016767  170124  170124  MOV     WORK,WORK1
010766  042767  177760  170116  BIC     #177760,WORK1
010774  022767  000013  170110  CMP     #13,WORK1
011002  001007                  BNE     BALD1X
011004  052767  000020  170076  BIS     #BIT4,WORK
```

```
011012  042767  000017  170070          BIC     #17,WORK
011020  000402                          BR      .+6
011022  005267  170062          BALD1X: INC     WORK
011026  005067  170060                  CLR     WORK1
011032  062767  000002  170030  BLAD1:  ADD     #2,SAVE
011040  062767  000002  170024          ADD     #2,SAV1
011046  000724                          BR      WRDCMP          ;COMPARE NEXT WORD
011050  000205                  ADAM:   RTS     %5              ;EXIT THIS ROUTINE
                                ;
```

```
011052  052767  001000  167762  WDERR:  BIS     #BIT9,FLAG      ;SET ERROR BIT
011060  005267  170002                  INC     ERCOUNT         ;+1 ERROR COUNT
011064  005767  167776                  TST     ERCOUNT         ;WHEN COUNT = 0 EXIT
011070  001767                          BEQ     ADAM            ;COUNT ZERO EXIT
011072  004567  002022                  JSR     %5,CONV         ;CONVERT WORD ADDR TO ASCII
011076  001110                          WORK
011100  013541                          MES1
011102  000006                          6
011104  004567  002010                  JSR     %5,CONV
011110  001112                          WORK1
011112  013647                          MES4
011114  000004                          4
011116  017767  167746  002054          MOV     @SAVE,ACNVX     ;FETCH GOOD DATA
011124  004567  001770                  JSR     %5,CONV         ;CONVERT GOOD DATA TO ASCII
011130  013200                          ACNVX
011132  013665                          MES5
011134  000006                          6
011136  017767  167730  002034          MOV     @SAV1,ACNVX     ;FETCH BAD DATA
011144  004567  001750                  JSR     %5,CONV         ;CONVERT TO ASCII
011150  013200                          ACNVX
011152  013705                          MES6
011154  000006                          6
011156  016767  167660  002014          MOV     FLAG,ACNVX      ;WHICH READ THE
011164  042767  177774  002006          BIC     #177774,ACNVX   ;ERROR OCCURRED ON
011172  004567  001722                  JSR     %5,CONV
011176  013200                          ACNVX
011200  013746                          MES13
011202  000001                          1
011204  104000                          EMT     +0              ;PRINT MESSAGE
011206  013416                          HED1
011210  013746                          MES13
011212  013541                          MES1
011214  013647                          MES4
011216  013665                          MES5
011220  013705                          MES6
011222  177777                          -1
011224  032777  002000  167546          BIT     #BIT10,@SR      ;HALT ON ERROR
011232  001401                          BEQ     .+4
011234  000000                          HALT                    ;YES HALT BIT 10 SET IN SR
011236  000634                          BR      WRDINC          ;GO COMPARE NEXT WORD
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
```

```
;
;
;
;
;
;
;
```

```
                                ;THE PROGRAM DETERMINES HOW MUCH MEMORY
                                ;IS ON THE SYSTEM
                                ;
011240  012777  000001  167546  EXTMEN: MOV     #BIT0,@RKCS     ;CLEAR THE DISK
011246  105777  167542          TSTB    @RKCS
011252  100375                  BPL     .-4
011254  012777  000340  167520          MOV     #340,@CSR       ;LOCK UP PRIORITY LEVELS
011262  012767  011334  166514          MOV     #MAXREF,4       ;SET UP I/O BUS TRAP
011270  012767  000340  166510          MOV     #340,6
011276  012767  017446  167564          MOV     #17446,SAVE     ;SET UP FOR 4K
011304  005777  167560          EXREF:  TST     @SAVE           ;REFERENCE MEMORY
011310  022767  177446  167552          CMP     #177446,SAVE    ;TEST FOR GREATER THAN 28K
011316  001002                  BNE     .+6             ;BRANCH IF LESS THAN 32K
011320  000000                  HALT
011322  000404                  BR      MAXREF          ;LAST REFERENCE MADE TO I/O REG.
011324  062767  020000  167536          ADD     #20000,SAVE     ;SET UP FOR NEXT MEMORY REF.
011332  000764                  BR      EXREF           ;GO REFERENCE MEMORY
                                ;
                                ;ENTER HERE WHEN I/O BUS ERROR OCCURS
                                ;
011334  012767  000006  166442  MAXREF: MOV     #6,4            ;RESTORE I/O BUS TRAP
011342  005067  166440          CLR     6
011346  162767  020000  167514          SUB     #20000,SAVE     ;
011354  162767  014536  167506          SUB     #OUTBUF,SAVE    ;SET UP NEW WORD COUNT
011362  006067  167502          ROR     SAVE
011366  016767  167476  167452          MOV     SAVE,WRDCT      ;
011374  042767  000001  167466          BIC     #BIT0,SAVE
011402  012767  014536  167476          MOV     #OUTBUF,INBUF
011410  066767  167454  167470          ADD     SAVE,INBUF
011416  006067  167424          ROR     WRDCT
011422  042767  000377  167416          BIC     #377,WRDCT
011430  016767  167412  167426          MOV     WRDCT,SWRDCT
011436  000205                  RTS     %5
                                ;BACKGROUND TEST FOR INTERRUPTS
                                ;
011440  012767  011526  166346  XWAIT:  MOV     #RTIX,14        ;SET UP TRACE TRAP
011446  005067  166344          CLR     16
011452  005067  167416          CLR     PASS            ;SET UP TIME BASE
011456  052777  000020  167316          BIS     #BIT4,@CSR      ;SET TRACE BIT
011464  005027  000000          CLR     #0
011470  005267  177772          XINCW:  INC     XINCW-2
011474  105767  177766          TSTB    XINCW-2
011500  100373                  BPL     XINCW
011502  005267  167366          INC     PASS
011506  001401                  BEQ     .+4
011510  000765                  BR      XINCW-4
                                ;REPORT BACKGROUND TEST TIMED OUT
011512  042777  000020  167262          BIC     #BIT4,@CSR      ;CLEAR TRACE BIT
011520  104001                  EMT+1
011522  013756                  TIMO
011524  000000                  HALT
                                ;
011526  000002          RTIX:   RTI
```

```
                        ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
                        ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
                        ;
                        ;WORD COUNT REGISTER
011530  017777  167244  167260  SELWC:  MOV     @SR,@RKWC               ;MOV SR INTO WORD COUNT REG
011536  000774                  SELWC:  BR      SELWC
                        ;
                        ;CURRENT ADDRESS REGISTER
011540  017777  167234  167252  SELCMA: MOV     @SR,@RKBA       ;MOV SR INTO CURRENT ADDR REG
011546  000774                  SELCMA: BR      SELCMA
                        ;
                        ;DISK ADDRESS REGISTER
011550  017777  167224  167244  SELDAR: MOV     @SR,@RKDA       ;MOV SR INTO DISK ADDR REG
011556  000774                  SELDAR: BR      SELDAR
                        ;
                        ;
                        ;
                        ;DISK CONTROL STATUS REGISTER
011560  012777  000340  167214  SELDCS: MOV     #340,@CSR       ;LOCK UP INTERRUPTS
011566  012777  177777  167222          MOV     #177777,@RKWC   ;SET WORD COUNT =1 WORD
011574  012777  014536  167216          MOV     #OUTBUF,@RKBA   ;SET UP CURRENT ADDRESS
011602  017777  167172  167204          MOV     @SR,@RKCS       ;MOV SR INTO CONTROL REG
011610  032777  000001  167162          BIT     #BIT0,@SR       ;IS FUNCTION BITS SET
011616  001760                  SELDCS: BEQ     SELDCS          ;FUNCTION BITS NOT SET
011620  105777  167170          DKBUSY: TSTB    @RKCS           ;TEST FOR DISK NOT READY
011624  100375                  DKBUSY: BPL     DKBUSY          ;DISK STILL READY
011626  000754                          BR      SELDCS          ;DISK NOT BUSY SELECT NEW CR
                        ;
                        ;
                        ;POWER FAIL TEST #1
                        ;WRITE THE DRIVE WITH A 12525252 PATTERN
                        ;THEN REPORT "OK"
                        ;THE START WRITING THE SAME PATTERN
                        ;WHEN THE POWER FAIL OCCURS ABORT THE TRANSFER
                        ;AND SET UP THE ENTRY POINT FOR RESTART
                        ;
                        ;
                        ;WHEN POWER IS UP READ THE COMPLETE DRIVE
                        ;THERE SHOULD BE ONE SECTOR WITH TEN ERRORS
                        ;
                        ;
011630  012706  001000  PFT1:   MOV     #1000,%6        ;SET UP THE STACK
011634  004767  000316          JSR     %7,POWFAL       ;WRITE 125252 ON THE DRIVE
011640  012777  000001  167146  PFWAT:  MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
011646  105777  167142          TSTB    @RKCS
011652  100375                  BPL     .-4
011654  005067  167172          CLR     DMA
011660  012767  011640  167212          MOV     #PFWAT,HRDER    ;SET UP FOR HARD ERROR
011666  012767  011754  166130          MOV     #DOWN,24        ;SET UP POWER FAIL VECTOR
011674  012767  000340  166124          MOV     #340,26
011702  012767  000200  167154          MOV     #200,SWRDCT
011710  016767  167150  167130          MOV     SWRDCT,WRDCT
011716  104503                  MYBYWR: WRITE   +100
011720  032777  001000  167052          BIT     #BIT9,@SR       ;HOW ARE WE WAITING FOR INTERRUPTS
```

```
011726  001002                          BNE     .+6
011730  000004                          IOT
011732  000404                          BR      .+12
011734  016777  167100  167040          MOV     PRIORITY,@CSR
011742  000001                          WAIT
011744  004767  175514                  JSR     %7,DISBUF       ;SET UP DRIVE DATA BUFFERS
011750  000762                          BR      MYBYWR
011752  000732                          BR      PFWAT           ;RECYCLE
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;POWER DOWN ROUTINE
                                ;WAIT FOR THE TRANSFER TO BE COMPLETE
                                ;
                                ;
011754  105777  167034          DOWN:   TSTB    @RKCS
011760  100375                          BPL     .-4
011762  012767  011772  166034          MOV     #UP,24          ;SET UP POWER FAIL VECTOR
011770  000000                          HALT
                                ;
                                ;
                                ;
011772  012767  011754  166024  UP:     MOV     #DOWN,24                ;SET UP FOR NEXT POWER FAIL
012000  012706  001000                  MOV     #1000,%6        ;SET UP THE STACK
012004  105777  167016                  TSTB    @RKDS           ;TEST FOR DRIVE READY
012010  100375                          BPL     .-4
012012  000167  000332                  JMP     UPCHK           ;GO CHECK DRIVE
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;POWER FAIL TEST #2
                                ;WRITE THE DRIVE WITH A PATTERN
                                ;REPORT "OK"
                                ;THEN READ THE DRIVE AND WAIT FOR POWER FAIL
                                ;AFTERPOWER IS UP CHECK THE
                                ;DRIVE FOR ERRORS
                                ;**********THER SHOULD BE NO ERRORS**************
                                ;
                                ;
                                ;
012016  012706  001000          PFT2:   MOV     #1000,%6        ;SET UP THE STACK
012022  004767  000130                  JSR     %7,POWFAL       ;WRITE THE PATTERN ON THE DRIVE
012026  005067  167020          PWRFL:  CLR     DMA
012032  012767  012026  167040          MOV     #PWRFL,HRDER    ;SET UP FOR HARD ERRORS
012040  012767  012116  165756          MOV     #PWRDN,24       ;SET UP POWER FAIL VECTOR
012046  012767  000340  165752          MOV     #340,26
012054  104507                  CHKDSK: WRCHECK +100
012056  032777  001000  166714          BIT     #BIT9,@SR       ;HOW ARE WE WAITING FOR INTERRUPS
012064  001002                          BNE     .+6
```

```
012066  000004                          IOT
012070  000404                          BR      .+12
012072  016777  166742  166702          MOV     PRIORITY,@CSR
012100  000001                          WAIT
012102  004767  175356                  JSR     %7,DISBUF       ;SET UP NEXT BUFFER
012106  000762                          BR      CHKDSK
012110  104001                          EMT     +1
012112  014530                          END
012114  000744                          BR      PWRFL
                                ;
                                ;
                                ;
                                ;ROUTINE TO ABORT TRANSFER  DURING POWER FAIL
                                ;
                                ;
                                ;
012116  012777  000001  166670  PWRDN:  MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
012124  105777  166664          TSTB    @RKCS
012130  100375                  BPL     .-4
012132  000000                  HALT
                                ;
                                ;
                                ;
                                ;POWER UP SEQUENCE
                                ;
012134  012767  012116  165662  PWRUP:  MOV     #PWRDN,24       ;SET UP FOR NEXT POWER FAIL
012142  012706  001000          MOV     #1000,%6        ;SET UP THE STACK
012146  105777  166654          TSTB    @RKDS           ;TEST FOR DRIVE READY
012152  100375                  BPL     .-4
012154  000724                  BR      PWRFL           ;GO CHECK TH DRIVE
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;ROUTINE TO WRITE THE DRIVE WITH THE 1252525 DATA PATTERN
                                ;
                                ;
                                ;
                                ;REPORT "OK" WHEN READY FOR POWER FAIL
                                ;
                                ;
012156  012777  000001  166630  POWFAL: MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
012164  105777  166624          TSTB    @RKCS
012170  100375                  BPL     .-4
012172  011667  166706          MOV     (6),PASSC       ;FETCH RETURN ADDRESS
012176  012706  001000          MOV     #1000,%6        ;RESET STACK
012202  012767  000020  166644  MOV     #20,PATNU       ;SET UP THE PATTERN NO.
012210  005067  166636          CLR     DMA
012214  012767  002000  166642  MOV     #2000,SWRDCT    ;SET UP THE WORD COUNT
012222  016767  166636  166616  MOV     SWRDCT,WRDCT
012230  004567  176142          JSR     %5,PASEL        ;GENERATE DATA BUFFER
012234  012767  014536  166614  MOV     #OUTBUF,BUF     ;SET UP CURRENT ADDRESS
012242  012767  012156  166630  MOV     #POWFAL,HRDER   ;SET UP FOR HARD ERRORS
012250  104503          WRDNW:  WRITE   +100
```

```
012252  032777  001000  166520          BIT     #BIT9,@SR       ;HOW ARE WE WAITING FOR INTERRUPTS
012260  001002                          BNE     .+6
012262  000004                          IOT
012264  000404                          BR      .+12
012266  016777  166546  166506          MOV     PRIORITY,@CSR
012274  000001                          WAIT
012276  104507                          WRCHECK +100
012300  032777  001000  166472          BIT     #BIT9,@SR       ;HOW ARE WE WAITING FOR INTERRUPTS
012306  001002                          BNE     .+6
012310  000004                          IOT
012312  000404                          BR      .+12
012314  016777  166520  166460          MOV     PRIORITY,@CSR
012322  000001                          WAIT
012324  004767  175134                  JSR     %7,DISBUF       ;SET UP NEXT BUFFER
012330  000747                          BR      WRDNW
012332  104001                          EMT     +1
012334  012342                          OK
012336  000177  166542                  JMP     @PASSC          ;EXIT DATA WRITTEN
                                ;
                                ;
                                ;
012342      045                 OK:     .ASCII  /%"OK"@/
012343      042
012344      117
012345      113
012346      042
012347      100

                                ;
                                ;
                                ;
                                ;ROUTINE TO CHECK DATA AFTER POWER FAIL
                                ;
                                ;
012350  012777  000001  166436  UPCHK:  MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
012356  105777  166432                  TSTB    @RKCS
012362  100375                        BPL     .-4
012364  005067  166462                  CLR     DMA
012370  012767  012350  166502          MOV     #UPCHK,HRDER    ;SET UP FOR HARD ERROR
012376  104507                  CHKDAT: WRCHECK +100
012400  032777  001000  166372          BIT     #BIT9,@SR       ;HOW ARE WE WAITING FOR INTERRUPTS
012406  001002                          BNE     .+6
012410  000004                          IOT
012412  000404                          BR      .+12
012414  016777  166420  166360          MOV     PRIORITY,@CSR
012422  000001                          WAIT
012424  004767  175034                  JSR     %7,DISBUF       ;SET UP DATA BUFFERS
012430  000762                          BR      CHKDAT
012432  104001                          EMT     +1
012434  014530                          END
012436  000167  177176                  JMP     PFWAT   ;GO WAIT FOR ANOTHER POWER FAIL
                                ;
                                ;
                                ;
                                ;
```

```
                                ;
                                ;THIS ROUTINE ENABLES THE OPERATOR TO SELECT A TRACK STATICLY
                                ;THE ROUTINE DOES A ONE WORD READ TO SELECT THE TRACK
                                ;THE OPERATOR MAY CHANGE THE SWITCH REGISTER AT ANY TIME
                                ;SET DRIVE AND CYLINDER NO.IN SR
                                ;
012442  012777  000001  166344  TKSEL: MOV     #BIT0,@RKCS
012450  105777  166340          TSTB    @RKCS           ;TEST FOR READY
012454  100375                  BPL     .-4
012456  017767  166316  166426          MOV     @SR,WORK1       ;FETCH SR
012464  016767  166422  166416          MOV     WORK1,WORK
012472  042767  174000  166410          BIC     #174000,WORK
012500  000241                  CLC
012502  006167  166402          ROL     WORK
012506  006167  166376          ROL     WORK
012512  006167  166372          ROL     WORK
012516  006167  166366          ROL     WORK
012522  006167  166362          ROL     WORK
012526  016777  166356  166266          MOV     WORK,@RKDA      ;DISK ADDRESS LOADED
012534  016777  166346  166256          MOV     INBUF,@RKBA     ;LOAD CURRENT ADDRESS
012542  012777  177777  166246          MOV     #177777,@RKWC   ;LOAD WORD COUNT
012550  012777  000011  166236          MOV     #11,@RKCS       ;GO AND READ
012556  105777  166232          TSTB    @RKCS           ;TEST FOR READY
012562  100375                  BPL     .-4
012564  032777  000100  166234  CTBUSY: BIT     #BIT6,@RKDS
012572  001774                  BEQ     CTBUSY          ;WAIT FOR CONTROL NOT READY
012574  026777  166312  166176  SRCHG:  CMP     WORK1,@SR
012602  001317                  BNE     TKSEL           ;SR HAS CHANGED
012604  000773                  BR      SRCHG           ;SR HAS NOT CHANGED
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
```

```
                                ;
                                ;ROUTINE TO REPORT ERROR COUNT AND CONTENTS OF ONE REGISTER
                                ;
012606  004567  000306       STAER:  JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
012612  001110                       WORK                    ;DATA TO BE CONVERTED
012614  013705                       MES6                    ;ADDRESS OF MESSAGE
012616  000006                       6
012620  004567  000274               JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
012624  001066                       ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
012626  013502                       HED5                    ;ADDRESS OF MESSAGE
012630  000003                       3
012632  104000                       EMT     +0              ;REPORT MESSAGE
012634  013500                       HED5A
012636  013502                       HED5
012640  013705                       MES6
012642  177777                       -1
012644  000205                       RTS     %5              ;EXIT ROUTINE
                                ;
                                ;ROUTINE TO REPORT ERROR COUNT AND THE CONTENTS OF TWO REGISTERS
                                ;
                                ;
                                ;
012646  004567  000246       STAER1: JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
012652  001110                       WORK                    ;DATA TO BE CONVERTED
012654  013705                       MES6                    ;ADDRESS OF MESSAGE
012656  000006                       6
012660  004567  000234               JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
012664  001112                       WORK1                   ;DATA TO BE CONVERT
012666  013665                       MES5                    ;ADDRESS OF MESSAGE
012670  000006                       6
012672  004567  000222               JSR     %5,CONV         ;CONVERT OCTAL TO ASCII
012676  001066                       ERCOUNT                 ;ERROR COUNT TO BE CONVERTED
012700  013502                       HED5                    ;ADDRESS OF MESSAGE
012702  000003                       3
012704  104000                       EMT     +0              ;REPORT MESSAGE
012706  013500                       HED5A
012710  013502                       HED5
012712  013665                       MES5
012714  013705                       MES6
012716  177777                       -1
012720  000205                       RTS     %5              ;EXIT ROUTINE
                                ;
                                ;
                                ;ROUTINE TO DECODE EMT CALLS
                                ;EMT+1=TYPE ONE LINE OF TEXT
                                ;EMT+0=TYPE A SERIES OF LINES
012722  011600               EMTRP:  MOV     (6),%0
012724  022740  104001               CMP     #EMT+1,-(0)     ;WAS THE CALL EMT+1
012730  001056                       BNE     TYPS            ;NO! TYPE A SERIES OF LINES OF TEXT
012732  000400                       BR      TYP             ;YES TYPE ONE LINE OF TEXT
```

```
                                ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
012734  011600                  TYP:    MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
012736  062716  000002                  ADD     #2,@%6          ;SET UP EXIT.
012742  011000                          MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0.
012744  112067  000114          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
012750  122767  000100  000106          CMPB    #100,TYPDAT     ;CHECK FOR "@" CHARACTER
012756  001001                          BNE     TYPC            ;CRANCH IF NOT "@".
012760  000002                          RTI                     ;TERMINATOR CHAR. DONE. EXIT.
012762  122767  000045  000074  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR "%".
012770  001422                          BEQ     TYPF            ;BRANCH IF "%".
012772  122767  000042  000064          CMPB    #42,TYPDAT      ;NOT "%". CHECK FOR "#".
013000  001423                          BEQ     TYPG            ;BRANCH IF "#"
013002  004767  000002                  JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
013006  000756                          BR      TYPA
013010  032777  040000  165762  TYPD:   BIT     #BIT14,@SR
013016  001006                          BNE     TYEXIT
013020  116777  000040  165756          MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
013026  105777  165756                  TSTB    @TPS            ;WAIT FOR DONE FLAG.
013032  100375                          BPL     .-4
013034  000207          TYEXIT: RTS     %7                      ;EXIT
013036  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
013044  004767  177740                  JSR     %7,TYPD         ;GO TYPE CHAR.
013050  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
013056  004767  177726                  JSR     %7,TYPD         ;GO TYPE CHAR.
013062  000730                          BR      TYPA
013064  000000          TYPDAT: 0
                                ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
013066  011600                  TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
013070  062716  000002                  ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
013074  011067  000014                  MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
013100  022767  177777  000006          CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
013106  001001                          BNE     TYPSA           ;BRANCH IF NOT TERMINATOR.
013110  000002                          RTI                     ;TERMINATOR. EXIT
013112  104001          TYPSA:  EMT     +1                      ;CALL ON TYP SUB TO TYPE MESSAGE
013114  000000          TYPSB:  0                               ;ADDRESS OF MESSAGE GOES HERE
013116  000763                  BR      TYPS                    ;GO PROCESS NEXT MESSAGE
                                ;
                                ;
                                ;
                                ;
                                ;OCTAL TO ASCII CONVERT ROUTINE
                                ;
                                ;ENTER ROUTINE AS FOLLOWS
                                ;JSR%5,CONV
                                ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED
                                ;ADDR BYTE=LSB OF WHERE ASCII IS GOING
                                ;ASCII#=THE NUMBER OF ASCII CHAR. TO BE CONVERTED
                                ;
                                ;
013120  013567  000054          CONV:   MOV     @(5)+,ACNVX     ;VALUE OF # TO BE CONVERTED
013124  012501                          MOV     (5)+,%1         ;ASCII ADDR
013126  012502                          MOV     (5)+,%2         ;# OF ASCII CHAR
013130  060201                          ADD     %2,%1
```

```
013132  016703  000042          ACVN:   MOV     ACNVX,%3
013136  042703  177770                  BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT OCTAL#
013142  062703  000060                  ADD     #60,%3          ;SET UP ASCII#
013146  110341                          MOVB    %3,-(1)         ;STORE ASCII CHAR
013150  042767  000007  000022          BIC     #7,ACNVX
013156  006067  000016                  ROR     ACNVX           ;ROTATE OCTAL#
013162  006067  000012                  ROR     ACNVX
013166  006067  000006                  ROR     ACNVX
013172  005302                          DEC     %2              ;-1 FROM ASCII CHAR COUNT
013174  001356                          BNE     ACVN
013176  000205                          RTS     %5              ;EXIT # CONVERTED
013200  000000                  ACNVX:  0                       ;WORK REGISTER
                                ;
                                ;
                                ;
                                ;
                                ;ROUTINE TO HANDLE KEYBOARD CONVERSATION
                                ;ROUTINE ACCEPTS NUM. OR ALPHA CHARACTERS
                                ;
013202  012767  000040  000204  NOCHA:  MOV     #40,TSTCH       ;SET UP FOR NUM. CHAR.
013210  000403                          BR      TYST
013212  012767  000100  000174  ALPHA:  MOV     #100,TSTCH      ;SET UP FOR ALPHA CHAR
013220  012777  000340  165554  TYST:   MOV     #340,@CSR       ;LOCK UP INTERRUPTS
013226  005067  000160                  CLR     TEXBUF          ;CLEAR TEXT BUFFER REG
013232  105777  165554          TSTFLG: TSTB    @TKS            ;CHECK FOR FLAG
013236  100375                          BPL     TSTFLG          ;
013240  017777  165542  165536          MOV     @TKB,@TPB       ;CHARACTER IN BUFFER
013246  105777  165536                  TSTB    @TPS            ;ECHO CHARACTER
013252  100375                          BPL     .-4
013254  022777  000377  165524          CMP     #377,@TKB       ;CHECK FOR RUB-OUT
013262  001014                          BNE     CKCH            ;EXIT IF NOT RUB-OUT
013264  104001                          EMT     -1
013266  013727                          MES8                    ;REPORT RUB-OUT ACKNOWLEDGED
013270  042767  000007  000114          BIC     #7,TEXBUF
013276  006067  000110                  ROR     TEXBUF
013302  006067  000104                  ROR     TEXBUF
013306  006067  000100                  ROR     TEXBUF
013312  000747                          BR      TSTFLG          ;GO WAIT FOR NEW CHAR.
013314  022777  000215  165464  CKCH:   CMP     #215,@TKB       ;CHECK FOR CARRIAGE RETURN
013322  001001                          BNE     .+4
013324  000207                          RTS     %7              ;EXIT DELIMITER TYPED
013326  036777  000062  165452          BIT     TSTCH,@TKB
013334  001003                          BNE     CHOK
013336  104001                          EMT     -1              ;REPORT QUESTION MARK
013340  013725                          MES7
013342  000733                          BR      TSTFLG          ;WAIT FOR CORRECT CHAR.
013344  017767  165436  165536  CHOK:   MOV     @TKB,WORK
013352  042767  177770  165530          BIC     #177770,WORK
013360  000241                          CLC
013362  006167  000024                  ROL     TEXBUF
013366  000241                          CLC
013370  006167  000016                  ROL     TEXBUF
013374  000241                          CLC
013376  006167  000010                  ROL     TEXBUF
013402  066767  165502  000002          ADD     WORK,TEXBUF     ;ADD CHARACTER
```

```
013410  000710                    BR      TSTFLG          ;WAIT FOR NEW CHARACTER
```

```
013412  000000                  TEXBUF: 0
013414  000000                  TSTCH:  0
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;ERROR MESSAGE HEADERS
                                ;

        013416                          .EVEN
013416     045                  HED1:   .ASCII  /%DATA ERROR @/
013417     104
013420     101
013421     124
013422     101
013423     040
013424     105
013425     122
013426     122
013427     117
013430     122
013431     040
013432     100

                                ;
                                ;
013433     045                  HED2:   .ASCII  /%STATUS ERROR @/
013434     123
013435     124
013436     101
013437     124
013440     125
013441     123
013442     040
013443     105
013444     122
013445     122
013446     117
013447     122
013450     040
013451     100

                                ;
                                ;
013452     045                  HED4:   .ASCII  /%DRIVE ADDRESS ERROR @/
013453     104
013454     122
013455     111
013456     126
013457     105
013460     040
013461     101
013462     104
013463     104
```

```
013464      122
013465      105
013466      123
013467      123
013470      040
013471      105
013472      122
013473      122
013474      117
013475      122
013476      040
013477      100

                                ;
                                ;
013500      045                 HED5A:  .ASCII  /%@/
013501      100

                                ;
                                ;
013502      040                 HED5:   .ASCII  /    ERROR COUNT @/
013503      040
013504      040
013505      040
013506      105
013507      122
013510      122
013511      117
013512      122
013513      040
013514      103
013515      117
013516      125
013517      116
013520      124
013521      040
013522      100

                                ;
                                ;
013523      045                 HED6:   .ASCII  /%RANEX ERROR @/
013524      122
013525      101
013526      116
013527      105
013530      130
013531      040
013532      105
013533      122
013534      122
013535      117
013536      122
013537      040
013540      100
```

```
                                ;
                                ;
                                ;MESSAGE TRAILERS
                                ;
                                ;
013541      040                 MES1:   .ASCII  /       RKDA @/
013542      040
013543      040
013544      040
013545      040
013546      040
013547      040
013550      122
013551      113
013552      104
013553      101
013554      040
013555      100

                                ;
                                ;
                                ;
013556      040                 MES1A:  .ASCII  /       RKER @/
013557      040
013560      040
013561      040
013562      040
013563      040
013564      040
013565      122
013566      113
013567      105
013570      122
013571      040
013572      100

                                ;
                                ;
                                ;
013573      040                 MES2:   .ASCII  /       RKCS @/
013574      040
013575      040
013576      040
013577      040
013600      040
013601      040
013602      122
013603      113
013604      103
013605      123
013606      040
013607      100

                                ;
013610      040                 MES2A:  .ASCII  /       RKDS @/
```

```
013611      040
013612      040
013613      040
013614      040
013615      040
013616      040
013617      122
013620      113
013621      104
013622      123
013623      040
013624      100

                                ;
013625      040                 MES3:   .ASCII  /       WORD COUNT@/
013626      040
013627      040
013630      040
013631      040
013632      040
013633      040
013634      127
013635      117
013636      122
013637      104
013640      040
013641      103
013642      117
013643      125
013644      116
013645      124
013646      100

                                ;
                                ;
                                ;
                                ;
013647      040                 MES4:   .ASCII  /     WRD ADR.@/
013650      040
013651      040
013652      040
013653      040
013654      127
013655      122
013656      104
013657      040
013660      101
013661      104
013662      122
013663      056
013664      100

                                ;
                                ;
                                ;
```

```
013665      040                     MES5:   .ASCII  /       GD DATA @/
013666      040
013667      040
013670      040
013671      040
013672      040
013673      040
013674      107
013675      104
013676      040
013677      104
013700      101
013701      124
013702      101
013703      040
013704      100
```

```
013705      040                 MES6:   .ASCII  /       BD DATA @/
013706      040
013707      040
013710      040
013711      040
013712      040
013713      040
013714      102
013715      104
013716      040
013717      104
013720      101
013721      124
013722      101
013723      040
013724      100

                                )
                                )
                                )
013725      077                 MES7:   .ASCII  /?@/
013726      100

                                )
                                )
                                )
                                )
013727      057                 MES8:   .ASCII  '/@'
013730      100

                                )
                                )
                                )
                                )
013731      045                 MES11:  .ASCII  /%UNIT NO.@/
013732      125
013733      116
013734      111
013735      124
013736      040
013737      116
013740      117
013741      056
013742      100

                                )
                                )
013743      040                 MES12:  .ASCII  /  @/
013744      040
013745      100

                                )
                                )
013746      040                 MES13:  .ASCII  /  READ @/
```

```
013747      040
013750      122
013751      105
013752      101
013753      104
013754      040
013755      100

                                    ;
                                    ;
013756      045                     TIMO:   .ASCII  /%CPU BKGRND TIMED OUT@/
013757      103
013760      120
013761      125
013762      040
013763      102
013764      113
013765      107
013766      122
013767      116
013770      104
013771      040
013772      124
013773      111
013774      115
013775      105
013776      104
013777      040
014000      117
014001      125
014002      124
014003      100
```

```
                                ;CONVERSATION TEXT
                                ;
                                ;
014004  040                     SPECMES:  .ASCII  /       SWRDCT@/
014005  040
014006  040
014007  040
014010  040
014011  040
014012  040
014013  123
014014  127
014015  122
014016  104
014017  103
014020  124
014021  100

                                ;;
014022  045                     CON1:   .ASCII  /%DATA TEST ONLY? (YES-NO)@/
014023  104
014024  101
014025  124
014026  101
014027  040
014030  124
014031  105
014032  123
014033  124
014034  040
014035  117
014036  116
014037  114
014040  131
014041  077
014042  040
014043  050
014044  131
014045  105
014046  123
014047  055
014050  116
014051  117
014052  051
014053  100

                                ;
                                ;
                                ;
014054  045                     CON2:   .ASCII  /%MULTI DRIVE MODE?(YES-NO)@/
014055  115
014056  125
014057  114
014060  124
```

```
014061      111
014062      040
014063      104
014064      122
014065      111
014066      126
014067      105
014070      040
014071      115
014072      117
014073      104
014074      105
014075      077
014076      050
014077      131
014100      105
014101      123
014102      055
014103      116
014104      117
014105      051
014106      100

                                ;
                                ;
014107      045                 CON3:   .ASCII  /%NUMBER OF DRIVES 1 TO 10 OCTAL?@/
014110      116
014111      125
014112      115
014113      102
014114      105
014115      122
014116      040
014117      117
014120      106
014121      040
014122      104
014123      122
014124      111
014125      126
014126      105
014127      123
014130      040
014131      061
014132      040
014133      124
014134      117
014135      040
014136      061
014137      060
014140      040
014141      117
014142      103
014143      124
014144      101
```

```
014145      114
014146      077
014147      100


                                    ;
                                    ;
014150      045                     CON4:   .ASCII  /%EXERCISE DRIVE?@/
014151      105
014152      130
014153      105
014154      122
014155      103
014156      111
014157      123
014160      105
014161      040
014162      104
014163      122
014164      111
014165      126
014166      105
014167      077
014170      100


                                    ;
                                    ;
014171      045                     CON5:   .ASCII  /%OPTIONAL WORD COUNT (YES-NO)@/
014172      117
014173      120
014174      124
014175      111
014176      117
014177      116
014200      101
014201      114
014202      040
014203      127
014204      117
014205      122
014206      104
014207      040
014210      103
014211      117
014212      125
014213      116
014214      124
014215      040
014216      050
014217      131
014220      105
014221      123
014222      055
014223      116
014224      117
014225      051
```

```
014226     100

                                ;
                                ;
014227     045                  CON6:   .ASCII  /%LENGTH (1 TO SWRDCT)?@/
014230     114
014231     105
014232     116
014233     107
014234     124
014235     110
014236     040
014237     050
014240     061
014241     040
014242     124
014243     117
014244     040
014245     123
014246     127
014247     122
014250     104
014251     103
014252     124
014253     051
014254     077
014255     100

                                ;
                                ;
014256     045                  CON7:   .ASCII  /%STARTING SURFACE AND SECTOR?@/
014257     123
014260     124
014261     101
014262     122
014263     124
014264     111
014265     116
014266     107
014267     040
014270     123
014271     125
014272     122
014273     106
014274     101
014275     103
014276     105
014277     040
014300     101
014301     116
014302     104
014303     040
014304     123
014305     105
014306     103
```

```
014307      124
014310      117
014311      122
014312      077
014313      100


                                ;
                                ;
014314      045                 CON7A:  .ASCII  /%SELECT CYLINDER FROM SR0-7? (YES-NO)@/
014315      123
014316      105
014317      114
014320      105
014321      103
014322      124
014323      040
014324      103
014325      131
014326      114
014327      111
014330      116
014331      104
014332      105
014333      122
014334      040
014335      106
014336      122
014337      117
014340      115
014341      040
014342      123
014343      122
014344      060
014345      055
014346      067
014347      077
014350      040
014351      050
014352      131
014353      105
014354      123
014355      055
014356      116
014357      117
014360      051
014361      100


                                ;
014362      045                 CON8:   .ASCII  /%OPTIONAL DATA PATTERN NO.?@/
014363      117
014364      120
014365      124
014366      111
014367      117
014370      116
```

```
014371      101
014372      114
014373      040
014374      104
014375      101
014376      124
014377      101
014400      040
014401      120
014402      101
014403      124
014404      124
014405      105
014406      122
014407      116
014410      040
014411      116
014412      117
014413      056
014414      077
014415      100

                                ;
                                ;
014416      045                 CON9:   ,ASCII  /%WRITE?(YES-NO)@/
014417      127
014420      122
014421      111
014422      124
014423      105
014424      077
014425      050
014426      131
014427      105
014430      123
014431      055
014432      116
014433      117
014434      051
014435      100

                                ;
                                ;
014436      045                 CON10:  ,ASCII  /%WRITE CHECK?(YES-NO)@/
014437      127
014440      122
014441      111
014442      124
014443      105
014444      040
014445      103
014446      110
014447      105
014450      103
014451      113
```

```
014452      077
014453      050
014454      131
014455      105
014456      123
014457      055
014460      116
014461      117
014462      051
014463      100

                                ;
                                ;
014464      045                 CON11:  .ASCII  /%READ?(YES-NO)@/
014465      122
014466      105
014467      101
014470      104
014471      077
014472      050
014473      131
014474      105
014475      123
014476      055
014477      116
014500      117
014501      051
014502      100

                                ;
014503      045                 CON12:  .ASCII  /%READ CHECK?(YES-NO)@/
014504      122
014505      105
014506      101
014507      104
014510      040
014511      103
014512      110
014513      105
014514      103
014515      113
014516      077
014517      050
014520      131
014521      105
014522      123
014523      055
014524      116
014525      117
014526      051
014527      100
```

```
                                    ;
014530      045                     END:    .ASCII  /%END@/
014531      105
014532      116
014533      104
014534      100

            014536                          .EVEN
```

```
011044  000000                  TEXBUF: 0
011046  000000                  TSTCH:  0
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;ERROR MESSAGE HEADERS
                                ;
        011050                          .EVEN
011050      045                 HED5A:  .ASCII  /%0/
011051      100

011052      040                 HED5:   .ASCII  /    ERROR COUNT 0/
011053      040
011054      040
011055      040
011056      105
011057      122
011060      122
011061      117
011062      122
011063      040
011064      103
011065      117
011066      125
011067      116
011070      124
011071      040
011072      100

                                ;
                                ;
                                ;MESSAGE TRAILERS
                                ;
                                ;
                                ;
                                ;
011073      040                 MES5:   .ASCII  /       GOOD DATA  0/
011074      040
011075      040
011076      040
011077      040
011100      040
011101      040
011102      107
011103      117
011104      117
011105      104
011106      040
011107      104
011110      101
011111      124
```

A2XSEK	003414	CLRSUR	010046	ERR20	005300	MES6	013705
ACNVX	013200	CMDAE	007700	ERR3	002600	MES7	013725
ACVN	013132	COMDAR	007614	ERR4	003030	MES8	013727
ADAM	011050	COMPAR	010632	ERR5	003112	MSTR	005742
ADEHR	002560	CON1	014022	ERR6	003212	MYBYWR	011716
ADHGT	002532	CON10	014436	ERR7	003536	NBOSL	004160
ADRD	005546	CON11	014464	ESH	005442	NOCHA	013202
ADRREP	007116	CON12	014503	ESRDCH	005650	NOISET	004440
ADT1	002204	CON2	014054	EXGEN	010562	OK	012342
ADT2	002634	CON3	014107	EXMFLG	002170	OPDAR	001646
ADT2UX	003270	CON4	014150	EXRAX	006404	OPDSEL	007400
ADT2X	003036	CON5	014171	EXREF	011304	OPPAT	001702
ADT3	005050	CON6	014227	EXTAD2	003264	OPRD	002010
ADT3UX	003774	CON7	014256	EXTDR	010160	OPWCK	002064
ADTST	002112	CON7A	014314	EXTMEN	011240	OPWRT	001762
AISRY	004300	CON8	014362	EXTPP	006510	OUTBUF	014536
AKH	007716	CON9	014416	FILDAT	010422	PASEL	010376
ALPHA	013212	CONM	001342	FLAG	001042	PASS	001074
ASKWC	001524	CONTIN	010010	HED1	013416	PASSC	001104
AU2LPA	003650	CONV	013120	HED2	013433	PAT0	010570
AUX2	003566	CSR	001002	HED4	013452	PAT1	010572
BALD1X	011022	CTBUSY	012564	HED5	013502	PAT10	010610
BIT0	000001	DATAT	005322	HED5A	013500	PAT11	010612
BIT1	000002	DATATX	005316	HED6	013523	PAT12	010614
BIT10	002000	DATTES	001464	HINUM	010566	PAT13	010616
BIT11	004000	DECBLK	007540	HRDER	001100	PAT14	010620
BIT12	010000	DELMES	006752	HRDEXI	007210	PAT15	010622
BIT13	020000	DISBUF	007464	INBUF	001106	PAT16	010624
BIT14	040000	DISK	006514	INCADT	002550	PAT17	010626
BIT15	100000	DKBUSY	011620	INCSEC	007510	PAT2	010574
BIT2	000004	DKINT	006636	INCSUR	007550	PAT20	010630
BIT3	000010	DMA	001052	INTEXT	007350	PAT3	010576
BIT4	000020	DOWN	011754	LCONM	001320	PAT4	010600
BIT5	000040	DSKDR	001416	LDAT	005364	PAT5	010602
BIT6	000100	DSKNOR	001076	LKADER	005264	PAT6	010604
BIT7	000200	DSKNOS	010162	LODEN	010252	PAT7	010606
BIT8	000400	DSKRD	005472	LONUM	010564	PATNU	001054
BIT9	001000	ELH	005532	LPAD2	003120	PFT1	011630
BLAD1	011032	EMTRP	012722	LPAD3	005306	PFT2	012016
BLNK	010676	END	014530	LPAU2X	003740	PFWAT	011640
BLOCK	001102	ERADR	007636	LPAU3X	004364	POLOK	003230
BLSZ	010232	ERCOUN	001066	MAXREF	011334	POWFAL	012156
BUF	001056	ERR1	002376	MES1	013541	PRIORI	001040
BUFEXI	007706	ERR10	003620	MES11	013731	PWAIT	005022
BUFINX	007734	ERR11	003724	MES12	013743	PWRDN	012116
CHKADT	002604	ERR12	004144	MES13	013746	PWRFL	012026
CHKDAT	012376	ERR13	004264	MES1A	013556	PWRUP	012134
CHKDSK	012054	ERR14	004542	MES2	013573	RANDOM	010436
CHKRA3	004234	ERR15	004642	MES2A	013610	RANER	006330
CHOK	013344	ERR16	004742	MES3	013625	RANEX	006016
CKCH	013314	ERR17	005160	MES4	013647	RANNU	001044
CKHRDE	007200	ERR2	002524	MES5	013665	RCHWAI	005640

RDCHK	002036	TKSR	001620
RDCK	104413	TPB	001004
RDCKLP	005600	TPS	001010
RDSECT	002450	TRACK	001050
RDWAIT	005522	TSRDY	002764
READ	104405	TSTAUX	003514
REDA	007004	TSTCH	013414
REPOEN	006476	TSTFLG	013232
RKBA	001020	TSTNXS	005166
RKCS	001014	TSTRA3	004114
RKDA	001022	TSTZER	005112
RKDB	001030	TWRDCT	001060
RKDS	001026	TYEXIT	013034
RKER	001024	TYP	012734
RKMR	001032	TYPA	012744
RKWC	001016	TYPC	012762
RORBLK	010316	TYPD	013010
RTIX	011526	TYPDAT	013064
SANHT	002542	TYPF	013036
SAV1	001072	TYPG	013050
SAVE	001070	TYPS	013066
SAW1	004560	TYPSA	013112
SAW2	004660	TYPSB	013114
SAW3	004760	TYST	013220
SAWTH	004376	UP	011772
SEABUF	002302	UPCHK	012350
SEEKCY	004054	VECTOR	001034
SELCMA	011540	WCCON	001544
SELDAR	011550	WCWAIT	005724
SELDCS	011560	WDERR	011052
SELWC	011530	WORK	001110
SETADT	002416	WORK1	001112
SETBLK	010260	WORK2	001114
SHIFT	010454	WRAUSK	003352
SLH	005656	WRCHEC	104407
SOFTER	007304	WRDCMP	010720
SPECME	014004	WRDCT	001046
SPECOR	001242	WRDINC	010730
SR	001000	WRDNW	012250
SRCHG	012574	WRITE	104403
STAER	012606	WRLG	006066
STAER1	012646	WRNEXB	002404
START	001116	WRWAIT	005424
STATUS	001036	XESH	005734
SWRDCT	001064	XINCSE	007762
SWSUR	007604	XINCSU	010020
TDMA	001062	XINCW	011470
TEXBUF	013412	XSEABU	002326
TIMQ	013756	XSLH	005434
TKB	001006	XWAIT	011440
TKS	001012	XWPAT	005042
TKSEL	012442		

ERRORS DETECTED: 0

RUN-TIME: 18 SECONDS

5K CORE USED

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRKG-A-D
PRODUCT NAME:	RK11 RANDOM EXERCISER
DATE CREATED:	JUNE 1, 1972
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	E. HAIGHT

COPYRIGHT © 1971, 1972
DIGITAL EQUIPMENT CORPORATION

1, ABSTRACT

THE RK11 RANDOM EXERCISER IS AN ATTEMPT TO SIMULATE A USERS ENVIRONMENT OF AN RK11 SYSTEM, THIS TEST IS NOT MEANT TO BE A DIAGNOSTIC, IF PROBLEMS ARE ENCOUNTERED WHILE RUNNING THIS PROGRAM, IT IS SUGGESTED THAT THE OPERATOR REVERT BACK TO THE RK11 DIAGNOSTIC (DISK DATA MAINDEC DZRKH),

2, REQUIREMENTS

2,1 EQUIPMENT

PDP-11 (MIN,4K)
TELETYPE
RK11 DISK CONTROL
RK (Ø2, Ø3, Ø4, Ø5) DISK DRIVE (MAX,8)

2,2 STORAGE

THE RANDOM EXERCISER OCCUPIES ALL OF MEMORY OF TO THE LOWER LIMIT OF THE ABSOLUTE LOADER,

3, LOADING PROCEDURE

3,1 METHOD OF LOADED RANDOM EXERCISER TAPE

PROGRAM FORMAT ABSOLUTE
A, VERIFY THE BOOT LOADER IS IN MEMORY,
B, SET SWITCH REGISTER EQUAL TO *50Ø',

MEMORY SIZE*	
4K	17
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

C, DEPRESS LOADED ADDRESS
D, DEPRESS START,

4, STARTING PROCEDURE

A, SET SWITCH REGISTER EQUAL TO 200',
B, DEPRESS LOAD ADDRESS,
C, SET SWITCH REGISTER EQUAL TO Ø',
D, DEPRESS START,

5. OPERATING PROCEDURE

AT THE START OF THE PROGRAM THE NUMBER OF DRIVES
THAT ARE ON LINE WILL BE REPORTED TO THE USER

XX DRIVE(S) ON LINE

5.1 CONTROL SWITCH SETTINGS

N/A

6. MESSAGE REPORTS

6.1 ERROR REPORTS

ERROR XXXXXXRKCS XXXXXXRKER XXXXXXRKDS XXXXXXRKDA

A B C D

A=CONTROL FUNCTION REGISTER
B=CONTROL ERROR REGISTER
C=DRIVE STATUS REGISTER
D=CONTROL ADDRESS REGISTER

BIT LAYOUT OF RKCS REGISTER

BIT15=ERROR
BIT14= HARD ERROR
BIT13= SEARCH COMPLETE
BIT12= MAINTENANCE ENABLE
BIT11= NOT USED
BIT10= FORMAT
BIT9= READ/WRITE ALL
BIT8 = STOP ON SOFT ERROR
BIT7 = READY
BIT6 = INTERRUPT ENABLE
BIT5 = EXTENDED MEMORY 1 (XM1)
BIT4 = EXTENDED MEMORY 0 (XM0)
BIT3-2-1= FUNCTION REGISTER
BIT0 = GO

BIT 3	BIT 2	BIT 1	OPERATION
0	0	0	RESET
0	1	0	READ
0	0	1	WRITE
0	1	1	WRITE CHECK
1	0	0	SEEK
1	0	1	READ CHECK
1	1	0	DRIVE RESET
1	1	1	WRITE LOCK

BIT LAYOUT OF RKER REGISTER

BIT15= DRIVE ERROR (DRE)
BIT14= OVERRUN (OVR)
BIT13= WRITE LOCK OUT VIOLATION (WLO)
BIT12= SEEK ERROR (SKE)
BIT11= PROGRAMMING ERROR (PCE)
BIT10= NON-EXISTENT MEMORY ERROR (NXM)
BIT9 = DATA LATE ERROR (DTL)
BIT8 = READ TIMING ERROR (DTL)
BIT7 = NON-EXISTENT DISK ERROR (NXD)
BIT6 = NON-EXISTENT CYLINDER ERROR (NXC)
BIT5 = NON-EXISTENT SECTOR (NXS)
BIT4-3-2= NOT USED
BIT1= CHECKSUM ERROR (CSE)
BIT0= WRITE CHECK ERROR (WCE)

XXXXXX RKDS

LAYOUT OF RKDS BITS

BIT15-14-13= IDENTIFICATION OF DRIVE
BIT12= DRIVE POWER LOW (DPL)
BIT11= HIGH DENSITY DRIVE (HDEN)
BIT10= DRIVE UNSAFE (DRU)
BIT9 = SEEK INCOMPLETE (SIN)
BIT8 = SECTOR COUNTER OK (SOK)
BIT7 = DRIVE READY (DRY)
BIT6 = ACCESS READY (ARDY)
BIT5 = WRITE PROTECT STATUS (WPS)
BIT4 = SECTOR COUNTER EQUALS SECTOR ADDRESS (SC=SA)
BIT3-0= SECTOR COUNTER (SC)

6,2 DRIVE ZERO OFF LINE

DRIVE ZERO OFF LINE

THIS MESSAGE WILL BE REPORTED IF DRIVE ZERO IS NOT
IN THE READY STATE.

6,3 ERROR HALTS

3104 ERROR WHILE WRITING DATA ON THE SURFACE
REGISTER 0 CONTAINS DATA WRITTEN (DEPRESS CONTINUE)

3112 REGISTER 0 CONTAINS RKCS
DEPRESS CONTINUE

3120 REGISTER 0 CONTAINS RKER
DEPRESS CONTINUE

3126 REGISTER 0 CONTAINS RKDS
DEPRESS CONTINUE

3200 ERROR WHILE READING FROM THE DISK
REGISTER 0 CONTAINS DATA TO BE READ (DEPRESS CONTINUE)

3206 REGISTER 0 CONTAINS RKCS
DEPRESS CONTINUE

3214 REGISTER 0 CONTAINS RKER
DEPRESS CONTINUE

3222 REGISTER 0 CONTAINS RKDS
DEPRESS CONTINUE

3240 REGISTER 0 CONTAINS DATA READ FROM THE DISK
DATA INCORRECT.

IF ONE OF THESE ERRORS OCCUR THE OPERATOR SHOULD
RELOAD DISK DATA AND USE THAT TO FIND HIS PROBLEM.

6.4 END

END

THIS MESSAGE IS REPORTED AT THE END OF EACH PASS OF THE PROGRAM.
A PASS EQUALS 25 ITERATIONS OF A RANDOM BUFFER.

7. MISCELLANEOUS

ALL DRIVES MUST BE PLACED IN SEQUENTIAL ORDER AND ON-LINE
RANGING FROM ZERO UP.

```
                        ;***RK11 RANDOM EXERCISER***
                        ;  THIS IS A RANDOM DATA, RANDOM CYLINDER, SECTOR ADDRESS,
                        ;  *RANDOM DRIVE NUMBER AND RANDOM WORD COUNT,
                        ;  AT THE START OF THE PROGRAM WILL
                        ;  REPORT THE NUMBER OF DRIVES ON LINE
                        ;  ON THE SYSTEM,
                        ;  THE PROGRAM WILL THEN RESTORE ALL THE DRIVES
                        ;  AND GENERATE A RANDOM BUFFER,
                        ;  THE FIRST FEW WORDS OF THE BUFFER WILL BE
                        ;  USED AS RANDOM PARMETERS FOR THE PROGRAM,
                        ;  THE PROGRAM THEN STARTS ALL THE DRIVES
                        ;  SEEKING, THE FIRST DRIVE COMPLETED SEEKING
                        ;  WILL BE THE FIRST DRIVE TO WRITE,
                        ;  AFTER EACH WRITE THE CONTROL WILL BE CHECKED
                        ;  FOR THE NEXT DRIVE WHICH HAS COMPLETE ITS'S
                        ;  SEEK AND THEN IT STARTS WRITING, AND SO ON
                        ;  UNTIL ALL DRIVES HAVE COMPLETED THEIR WRITES,
                        ;  THEN ALL OF THE DRIVES START SEEKING TO
                        ;  A NEW RANDOM ADDRESS, WHEN THIS SEEK IS
                        ;  COMPLETED THE FIRST DRIVE DONE IS TOLD TO
                        ;  WRITE CHECK THE DATA WRITTEN, WHEN ALL
                        ;  DRIVE HAVE COMPLETED THIS CYCLE A NEW RANDOM
                        ;  BUFFER IS GENERATE AND THE SEQUENCE
                        ;  STARTS OVER AGAIN.
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
000001                  BIT0=   1
000002                  BIT1=   2
000004                  BIT2=   4
000010                  BIT3=   10
000020                  BIT4=   20
000040                  BIT5=   40
000100                  BIT6=   100
000200                  BIT7=   200
000400                  BIT8=   400
001000                  BIT9=   1000
002000                  BIT10=  2000
004000                  BIT11=  4000
010000                  BIT12=  10000
020000                  BIT13=  20000
040000                  BIT14=  40000
100000                  BIT15=  100000
                        ;
                        ;
                        ;
000503                  WRITE=  503
000507                  WRCHECK= 507
000111                  SEEK=   111
000240                  NOP=    240
```

```
                000042                  .=42
000042  000000                 0
                        ;
                000200                  .=200
000200  000167  000064         JMP     START                   ;GO TO START OF TEST
                001000                  .=1000
                001000                  .EVEN
                        ;
                        ;I/O ADDRESSES
001000  177570          SR:     177570                  ;SWITCH REGISTER
001002  177776          CSR:    177776                  ;PROCESSOR STATUS REGISTER
001004  177566          TPB:    177566                  ;TELETYPE REGISTERS
001006  177562          TKB:    177562
001010  177564          TPS:    177564
001012  177560          TKS:    177560
                        ;
                        ;
                        ;DISK REGISTERS
001014  177404          RKCS:   177404                  ;CONTROL STATUS REGISTER
001016  177406          RKWC:   177406                  ;WORD COUNT REGISTER
001020  177410          RKBA:   177410                  ;CURRENT ADDRESS REGISTER
001022  177412          RKDA:   177412                  ;DISK ADDRESS REGISTER
001024  177402          RKER:   177402                  ;DISK ERROR REGISTER
001026  177400          RKDS:   177400                  ;DRIVE STATUS REGISTER
001030  000220          VECTOR: 220                     ;INTERRUPT VECTOR ADDRESS
001032  000222          STATUS: 222
001034  000200          PRIORITY:BIT7                   ;CONTROL PRIORITY LEVEL
                        ;
                        ;
001036  000000          WCSAV:  0                       ;WORD COUNT LOADED
001040  004304          BASAV:  OUTBUF                  ;BUSS ADDRESS LOADED
001042  000000          PASS:   0                       ;PASS COUNTER
001044  000000          COUNT:  0
001046  000000          DRIVE:  0
001050  000000          SAVE:   0
001052  000000          BLOCK:  0
001054  000000          WORK1:  0
001056  000000          WORK:   0
001060  000000          MEMEND: 0
001062  000000          NEWFUN: 0
001064  000000          DATA:   0
001066  000000          ADDR:   0
```

```
001070  000005                  START:  RESET                   ;CLEAR THE WORLD
001072  012706  001000                  MOV     #1000,%6        ;SET UP STACK
001076  004567  002310                  JSR     %5,EXTMEM       ;DETERMINE SIZE OF MEMORY
001102  012706  001000                  MOV     #1000,%6        ;RESET STACK
001106  012767  003746  176710          MOV     #DOWN,24        ;SET UP POWER FAIL VECTOR
001114  012767  000340  176704          MOV     #340,26
001122  012767  003564  176700          MOV     #EMTRP,30       ;SET UP TTY TRAP
001130  012767  000340  176674          MOV     #340,32
001136  012777  000340  177636          MOV     #340,@CSR       ;LOCK PROCESSOR
001144  012777  002122  177636          MOV     #INTHN,@VECTOR  ;SET UP THE INTERRUPT VECTOR
001152  012777  000340  177652          MOV     #340,@STATUS
001160  005077  177636                  CLR     @RKDA
001164  005067  002702                  CLR     TEXBUF
001170  105777  177632          CONM:   TSTB    @RKDS           ;IS THE DRIVE READY
001174  100013                          BPL     RKNODV          ;DRIVE NONEXISTENT
001176  062767  000001  002666          ADD     #1,TEXBUF
001204  022767  000011  002660          CMP     #11,TEXBUF      ;DO WE HAVE 8 DRIVES ON THE SYSTEM
001212  001404                          BEQ     RKNODV          ;YES EXIT
001214  062777  020000  177600          ADD     #20000,@RKDA    ;SET UP FOR NEXT DRIVE
001222  000762                          BR      CONM            ;TEST NEXT DRIVE
001224  005767  002642          RKNODV: TST     TEXBUF
001230  001004                          BNE     REPDV           ;REPORT DRIVE NUMBERS
001232  104001                          EMT     +1
001234  004256                          NODRV                   ;REPORT NO DRIVES PRESENT
001236  000000                          HALT
001240  000777                          BR      .
001242  004567  002542          REPDV:  JSR     %5,CONV         ;REPORT THE NUMBER OF DRIVES
001246  004072                          TEXBUF
001250  004074                          CON1
001252  000001                          1
001254  104000                          EMT     +0
001256  004216                          MESS
001260  004074                          CON1
001262  177777                          -1
001264  005467  002602                  NEG     TEXBUF
001270  012777  000001  177516  RESTART:MOV     #BIT0,@RKCS     ;CLEAR THE CONTROL
001276  105777  177512                  TSTB    @RKCS           ;WAIT FOR CONTROL READY
001302  100375                          BPL     .-4
                                ;
                                ;
                                ;
                                ;
                                ;ROUTINE TO RESTORE DRIVES AND LEAVE ROUTINE
                                ;WITH ALL HEADS MOVING;
                                ;
001304  016767  002562  177532  RESTORE: MOV    TEXBUF,COUNT
001312  105777  177510                  TSTB    @RKDS           ;TEST FOR DRIVE OK!
001316  100375                          BPL     .-4
001320  012777  000015  177466          MOV     #15,@RKCS       ;RESTORE DRIVE COMMAND
001326  105777  177462                  TSTB    @RKCS           ;IS CONTROL READY
001332  100375                          BPL     .-4
001334  032777  000100  177464          BIT     #BIT6,@RKDS     ;ARE THE HEADS STILL MOVING
001342  001774                          BEQ     .-6             ;TEST FOR ACCESS RADY
001344  005777  177444                  TST     @RKCS           ;WAS THERE AN ERROR IN THE RESTORE
```

```
001350  100002                          BPL     .+6             ;BRANCH IF THERE WAS NO ERROR
001352  000167  000612                  JMP     ERROR           ;REPORT THE ERROR OCCURRED
001356  005267  177462                  INC     COUNT           ;+1  COUNTER
001342  001404                          BEQ     DTMOD           ;YES!  EXIT
001364  062777  020000  177430          ADD     #20000,@RKDA    ;SETUP NEXT DRIVE #
001372  000747                          BR      RESTORE+6       ;RESTORE NEXT CONTROL
                                )
                                )
                                ;THIS ROUTINE IS USED TO VERIFY THE DATA LINES
                                ;WRITE ZEROES ON SURFACE ZERO (ONE WORD)
                                ;READ DATA AND VERIFY NO BITS ARE PICKED UP
                                ;WRITE ONES ON SURFACE ONE (ONE WORD)
                                ;READ DATA AND MAKE SURE NO BITS ARE DROPPED
                                )
                                )
                                )
001374  005067  177464          DTMOD:  CLR     DATA            ;SET DATA WORD TO ZERO
001400  005067  177442                  CLR     ADDR            ;SET DRIVE ADDRESS TO ZERO
001404  004767  001416                  JSR     %7,LINECK       ;GO WRITE THE DATA
001410  012767  177777  177446          MOV     #177777,DATA    ;SET DATA WORD TO ONE
001416  012767  000020  177442          MOV     #20,ADDR        ;SET ADDRESS TO SURFACE ONE
001424  004767  001376                  JSR     %7,LINECK       ;GO WRITE DATA
                                )
```

```
001430  012767  000031  177404  LETSGO: MOV     #25.,PASS       ;SET UP PASS COUNTER
001436  004567  001606          JSR     %5,GENRAN       ;GENERATE RANDOM BUFFER
001442  012777  000340  177332  LETSEEK: MOV    #340,@CSR       ;LOCK CPU PRIORITY
001450  005000                  CLR     %0
001452  016767  002414  177364          MOV     TEXBUF,COUNT    ;SET UP TO COUNT DRIVES
001460  012777  000001  177326          MOV     #BIT0,@RKCS     ;CLEAR CONTROL
001466  105777  177322                  TSTB    @RKCS           ;IS THE CONTROL READY
001472  100375                          BPL     .-4
001474  016002  004304          NEWSK:  MOV     OUTBUF(0),%2    ;FETCH RANDOM DRIVE ADDRESS
001500  004567  000664                  JSR     %5,ADTST        ;TEST ADDRESS LIMITS
001504  012777  000111  177302          MOV     #SEEK,@RKCS     ;GO SEEK
001512  105777  177276                  TSTB    @RKCS           ;IS DRIVE RDY FOR NEXT SEEK
001516  100375                          BPL     .-4
001520  062700  000002                  ADD     #2,%0           ;SET UP TO FETCH NEXT DRIVE ADDRESS
001524  005267  177314                  INC     COUNT
001530  001404                          BEQ     SKOVER          ;SEEKS OVER GET OUT
001532  062777  020000  177262          ADD     #20000,@RKDA    ;SET UP FOR NEXT DRIVE
001540  000755                          BR      NEWSK           ;DO NEXT SEEK
001542  016767  002324  177274  SKOVER: MOV     TEXBUF,COUNT    ;WAIT FOR DRIVE RDY
001550  012767  001564  177304          MOV     #LETSWR,NEWFUN
001556  005077  177220                  CLR     @CSR            ;CLEAR PRIORITY
001562  000777                          BR      .
001564  012777  000340  177210  LETSWR: MOV     #340,@CSR       ;LOCK UP CPU
001572  017700  177230                  MOV     @RKDS,%0
001576  000241                          CLC                     ;POL INTERRUPT OCCURRED FETCH DRIVE ID
001600  006100                          ROL     %0              ;SET UP DRIVE ID TO FETCH FORM TABLE OF RANDOM NUMBERS
001602  006100                          ROL     %0
001604  006100                          ROL     %0
001606  006100                          ROL     %0
001610  006100                          ROL     %0
001612  042700  177760                  BIC     #177760,%0
001616  016002  004314                  MOV     OUTBUF+10(0),%2 ;FETCH NEW DRIVE ADDRESS
001622  004567  000542                  JSR     %5,ADTST        ;IS ADDRESS OK
001626  016002  004324                  MOV     OUTBUF+20(0),%2;FETCH WORD COUNT
001632  004567  000642                  JSR     %5,BLSZ         ;IS WORD COUNT IN LIMITS
001636  012777  000503  177150          MOV     #WRITE,@RKCS    ;WRITE RANDOM DATA
001644  005077  177132                  CLR     @CSR
001650  000777                          BR      .               ;WAIT FOR INTERRUPTS

                                ;
                                ;THIS ROUTINE SETS DRIVES SEEKING
                                ;AFTER ALL THE WRITES HAVE BEEN COMPLETED

001652  012777  000340  177122  SECSEEK: MOV    #340,@CSR       ;LOCK UP CPU
001660  005000                  CLR     %0
001662  016767  002204  177154          MOV     TEXBUF,COUNT
001670  012777  000001  177116          MOV     #BIT0,@RKCS     ;CLEAR CONTROL
001676  105777  177112                  TSTB    @RKCS           ;WAIT FOR CONTROL READY
001702  100375                          BPL     .-4
001704  016002  004404          WATSEK: MOV     OUTBUF+100(0),%2;FETCH ADDRESS
001710  004567  000454                  JSR     %5,ADTST        ;IS ADDRESS OK
001714  012777  000111  177072          MOV     #SEEK,@RKCS     ;GO SEEK
001722  105777  177066                  TSTB    @RKCS           ;IS CONTROL READY
001726  100375                          BPL     .-4
```

```
001730  062700  000002                  ADD     #2,%0           ;SET UP NEXT DRIVE
001734  005267  177104                  INC     COUNT           ;ARE ALL DRIVES SEEKING
001740  001404                          BEQ     SECOVER         ;BRANCH IF ALL DRIVE SEEKING
001742  062777  020000  177052          ADD     #20000,@RKDA    ;SET UP NEXT DRIVE NUMBER
001750  000755                          BR      WATSEK
001752  016767  002114  177064  SECOVER:MOV     TEXBUF,COUNT
001760  012767  001774  177074          MOV     #CHKDAT,NEWFUN  ;SETUP TO WRITE CHECK NEXT
001766  005077  177010                  CLR     @CSR            ;CLEAR PRIORITY
001772  000777                          BR      .
```

```
                                ;
                                ;WRITE CHECK DATA WRITTEN ON FIRST SEEK
                                ;
001774  012777  000340  177000  CHKDAT: MOV     #340,@CSR       ;LOCK UP CPU
002002  017700  177020          MOV     @RKDS,%0        ;FETCH DRIVE NUMBER
002006  000241                  CLC                     ;POL INTERRUPT OCCURRED FETCH DRIVE ID
002010  006100                  ROL     %0              ;SET UP DRIVE ID TO FETCH FROM RANDOM TABLE
002012  006100                  ROL     %0
002014  006100                  ROL     %0
002016  006100                  ROL     %0
002020  006100                  ROL     %0
002022  042700  177760          BIC     #177760,%0
002026  016002  004314          MOV     OUTBUF+10(0),%2
002032  004567  000332          JSR     %5,ADTST        ;IS ADDRESS WITHIN BOUNDS
002036  016002  004324          MOV     OUTBUF+20(0),%2 ;FETCH WORD COUNT
002042  004567  000432          JSR     %5,BLSZ         ;IS WORD COUNT IN LIMIT
002046  012777  000507  176740  MOV     #WRCHECK,@RKCS
002054  005077  176722          CLR     @CSR
002060  000777                  BR      .               ;WAIT FOR INTERRUPTS
002062  005367  176754          DEC     PASS            ;DEC. PASS COUNTER
002066  001402                  BEQ     .+6             ;
002070  000167  177346          JMP     LETSEEK         ;RECYCLE
002074  013700  000042          MOV     @#42,%0
002100  001404                  BEQ     REPEND
002102  004710                  LOGICAL: JSR    %7,(0)
002104  000240                  NOP
002106  000240                  NOP
002110  000240                  NOP
002112  104001                  REPEND: EMT     +1
002114  004251                  END
002116  000167  177306          JMP     LETSGO          ;GO GENERATE A NEW RANDOM BUFFER
                                ;
                                ;INTERRUPT HANDLER FOR RANDOM EXERCISER
                                ;
                                ;
                                ;
002122  005777  176666          INTHN:  TST     @RKCS           ;WAS INT CAUSED BY ERROR
002126  100420                  BMI     ERROR           ;BRANCH ERROR OCCURRED
002130  032777  020000  176656  BIT     #BIT13,@RKCS    ;WAS INT CAUSED BY SEARCH COMP;
002136  001111                  BNE     EXIT            ;SEEK COMPLETE EXIT;
                                ;
                                ;
                                ;
002140  032777  000010  176646  RKDONE: BIT     #BIT3,@RKCS     ;ARE WE SEEKING OR RESTORING
002146  001401                  BEQ     .+4             ;NO; WE ARE WRITING OR WRITE/CHECKING
002150  000002                  RTI                     ;YES GO BACK AND WAIT
002152  005267  176666          INC     COUNT           ;+1 DONE COUNT
002156  001401                  BEQ     .+4             ;IF ALL DRIVES DONE EXIT
002160  000002                  RTI                     ;NOT ALL DRIVES ARE DONE
002162  062716  000002          ADD     #2,@#6          ;ALTER STACK
002166  000002                  RTI                     ;EXIT ALL DRIVES DONE
```

```
                                        ;
                                        ;ERROR REPORTING ROUTINE FOR INTERRUPT HANDLER
                                        ;
002170  017767  176620  176660  ERROR:  MOV     @RKCS,WORK      ;CONTROL FUNCTION REG
002176  004567  001606                  JSR     %5,CONV         ;CONVERT REG TO ASCII
002202  001056                          WORK
002204  004164                          MES2
002206  000006                          6
002210  017767  176612  176640          MOV     @RKDS,WORK      ;DRIVE STATUS REG
002216  004567  001566                  JSR     %5,CONV         ;CONVERT REG TO ASCII
002222  001056                          WORK
002224  004201                          MES2A
002226  000006                          6
002230  017767  176570  176620          MOV     @RKER,WORK      ;CONTROL ERROR REG;
002236  004567  001546                  JSR     %5,CONV         ;CONVERT REG TO ASCII
002242  001056                          WORK
002244  004147                          MES1A
002246  000006                          6
002250  017767  176546  176600          MOV     @RKDA,WORK      ;DRIVE ADDRESS REG;
002256  004567  001526                  JSR     %5,CONV         ;CONVERT REG TO ASCII
002262  001056                          WORK
002264  004132                          MES1                    ;REPORT THE STATUS OF CONTROL REGISTERS
002266  000006                          6
002270  104000                          EMT     +0
002272  004120                          MED2
002274  004164                          MES2
002276  004147                          MES1A
002300  004201                          MES2A
002302  004132                          MES1
002304  177777                          -1
002306  017700  176506                  MOV     @RKBA,%0        ;MOVE CURRENT ADDRESS TO THE DATA LIGHTS
002312  032777  100000  176460          BIT     #BIT15,@SR      ;TEST TO HALT ON ERROR
002320  100001                          BPL     .+4             ;BRANCH IF BIT15 NOT SET
002322  000000                          HALT                    ;HALT BIT15 SET
002324  032777  040000  176462          BIT     #BIT14,@RKCS    ;WAS IT A HARD ERROR
002332  001404                          BEQ     SOFTER          ;SOFT ERROR GO ON
002334  012706  001000                  MOV     #1000,%6        ;RESET STACK
002340  000167  176724                  JMP     RESTART         ;HARD ERROR RESTART TEST
002344  005777  176446          SOFTER: TST     @RKWC           ;IS THE WORD COUNT ZERO
002350  001404                          BEQ     EXIT            ;WORD COUNT IS ZERO EXIT
002352  052777  000001  176434          BIS     #BIT0,@RKCS     ;ISSUE GO
002360  000002                          RTI                     ;RETURN
002362  022626                  EXIT:   CMP     (6)+,(6)+       ;RESET THE STACK
002364  000177  176472                  JMP     @NEWFUN         ;RETURN AND WAIT FOR INT
                                        ;
                                        ;ROUTINE TO TEST IS DRIVE ADDRESS IS IN LIMITS
                                        ;
002370  010267  176462          ADTST:  MOV     %2,WORK         ;TEST FOR SECTOR LESS THAN 13
002374  042767  177760  176454          BIC     #177760,WORK
002402  022767  000014  176446          CMP     #14,WORK        ;IS SECTOR NO, GREATER THAN 13
002410  101002                          BHI     CYTST           ;SECTOR NO, OK
002412  042702  000010                  BIC     #BIT3,%2
002416  042702  160000          CYTST:  BIC     #160000,%2
002422  022702  014500                  CMP     #14500,%2       ;IS CYLINDER = TO OR LESS THAN 312
```

```
002426  101002                  BHI     ADROK           ;BRANCH IF CYLINDER OK
002430  042702  017400           BIC     #17400,X2       ;ALTER CYLINDER ADDRESS
```

```
002434  010067  176416          ADROK:  MOV     %0,WORK
002440  000241                  CLC
002442  006067  176410                  ROR     WORK            ;DRIVE # IN WORK LSB
002446  006067  176404                  ROR     WORK
002452  006067  176400                  ROR     WORK
002456  006067  176374                  ROR     WORK
002462  006067  176370                  ROR     WORK
002466  056702  176344                  BIS     WORK,%2         ;DRIVE # NOW IN %2
002472  010277  176324                  MOV     %2,@RKDA
002476  000205                          RTS     %5

                                ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
                                ;
002500  032777  004000  176320  BLG2I:  BIT     #BIT11,@RKDS    ;CHECK DENSITY
002506  001404                  BEQ     LODEN           ;DRIVE IS LOW DENSITY
002510  012767  000377  176334          MOV     #377,BLOCK      ;DRIVE IS HI DENSITY
002516  000403                  BR      SETBLK          ;SET UP NO. OF WORDS PER BLOCK
002520  012767  000177  176324  LODEN:  MOV     #177,BLOCK      ;SET UP BLOCK OVERFLOW
002526  026702  176316          SETBLK: CMP     SAVE,%2
002532  101002                  BHI     .+6             ;BRANCH IF WORD COUNT OK
002534  016702  176310                  MOV     SAVE,%2         ;FUDGE WORD COUNT
002540  036702  176306                  BIT     BLOCK,%2
002544  001406                  BEQ     RORBLK
002546  046702  176300                  BIC     BLOCK,%2
002552  005267  176274                  INC     BLOCK
002556  066702  176270                  ADD     BLOCK,%2
002562  006002                  RORBLK: ROR     %2
002564  006002                          ROR     %2
002566  006002                          ROR     %2
002570  006002                          ROR     %2
002572  006002                          ROR     %2
002574  006002                          ROR     %2
002576  006002                          ROR     %2
002600  032777  004000  176220          BIT     #BIT11,@RKDS    ;TEST FOR HI DENSITY
002606  001401                  BEQ     .+4             ;LOW DENSITY
002610  006002                  ROR     %2
002612  010267  176234                  MOV     %2,BLOCK        ;BLOCK COUNT
002616  017767  176200  176232          MOV     @RKDA,WORK      ;FETCH DRIVE ADDRESS
002624  042767  160000  176224          BIC     #160000,WORK    ;MASK OUT DRIVE #
002632  005002                  CLR     %2              ;SET UP POSITIVE BLOCK COUNT
002634  005202                  BLKCNT: INC     %2
002636  005367  176210                  DEC     BLOCK           ;# OF BLOCK IN RANDOM WORD COUNT
002642  001443                  BEQ     ADREXIT         ;EXIT: BLOCK COUNT OVERFLOWED
002644  016703  176206                  MOV     WORK,%3
002650  042703  177760                  BIC     #177760,%3      ;MASK ALL BUT SECTOR NUMBER
002654  022703  000013                  CMP     #13,%3          ;ARE WE ON THE LAST SECTOR
002660  001403                  BEQ     CHKSUR          ;YES: WERE ON LAST SECTOR
002662  005267  176170                  INC     WORK            ;INC DRIVE ADDRESS
002666  000762                  BR      BLKCNT          ;RECYCLE BLOCK COUNT
002670  042767  000017  176160  CHKSUR: BIC     #17,WORK        ;CLEAR SECTOR COUNT
002676  032767  000020  176152          BIT     #BIT4,WORK      ;WHICH SURFACE ARE ON
002704  001004                  BNE     CHKCYL          ;CHECK CYLINDER, WERE ON SURFACE 1
002706  052767  000020  176142          BIS     #BIT4,WORK      ;SET SURFACE TO ONE
002714  000747                  BR      BLKCNT          ;RECYCLE BLOCK COUNT
```

```
002716  042767  000020  176132  CHKCYL: BIC     #BIT4,WORK      ;CLEAR SURFACE BIT
002724  016703  176126          MOV     WORK,R3         ;FETCH DRIVE ADDRESS
002730  042703  160037          BIC     #160037,R3      ;MASK OUT ALL BUT CYLINDER #
002734  022703  014500          CMP     #14500,R3       ;IS IT THE LAST CYLINDER
002740  001404                  BEQ     ADREXIT         ;YES EXIT;
002742  062767  000040  176106          ADD     #40,WORK        ;SET UP NEXT CYLINDER #
002750  000731                  BR      BLKCNT          ;RECYCLE BLOCK COUNT
                        ;
                        ;CONVERT POSITIVE BLOCK TO WORD COUNT
                        ;
002752  000241          ADREXIT: CLC
002754  006102                  ROL     R2
002756  006102                  ROL     R2
002760  006102                  ROL     R2
002762  006102                  ROL     R2
002764  006102                  ROL     R2
002766  006102                  ROL     R2
002770  006102                  ROL     R2
002772  032777  004000  176026          BIT     #BIT11,@RKDS    ;IS IT A HI  DENSITY DRIVE
003000  001401                  BEQ     .+4             ;BRANCH IS NOT HI
003002  006102                  ROL     R2
003004  005402                  NEG     R2              ;CONVERT WORD COUNT TO TWO'S COMP;
003006  010277  176004          MOV     R2,@RKWC        ;LOAD WORD COUNT
003012  010267  176020          MOV     R2,WCSAV        ;SAVE THE WORD COUNT LOADED
                        ;       CMP     #OUTBUF,BASAV   ;DOES ADRESS FALL WITH-IN BUFFER LIMITS
                        ;       BHI     .+12            ;ADDRES OK
                        ;       MOV     #OUTBUF,BASAV   ;ADDRESS DID NOT FALL IN BUFFER LIMITS
                        ;       BR      LOADBA          ;LOAD THE BUSS ADDRESS
                        ;       MOV     BASAV,WORK
                        ;       ADD     R2,WORK
                        ;       CMP     WORK,MEMEND     ;WILL TRANSFER FIT IN MEMORY
                        ;       BLOS    LOADBA          ;ADDRESS OK
                        ;       MOV     #OUTBUF,BASAV
003016  016777  176016  175774  LOADBA: MOV     BASAV,@RKBA     ;LOAD CURRENT ADDRESS;
003024  000205                  RTS     R5              ;EXIT
                        ;
                        ;
                        ;
                        ;RK11 HANDLER TO WRITE DATA ON SURFACE ZERO AND ONE
                        ;ON ERROR REGISTER ZERO (DATA LIGHTS) CONTAINS DATA TO BE
                        ;WRITTEN ON THE SURFACE
                        ;
                        ;
003026  012777  177777  175762  LINECK: MOV     #-1,@RKWC       ;LOAD THE WORD COUNT
003034  012777  004004  175756          MOV     #OUTBUF,@RKBA   ;LOAD THE BUS ADDRESS
003042  016767  176016  001234          MOV     DATA,OUTBUF     ;LOAD THE DATA BUFFER
003050  016700  176010          MOV     DATA,R0         ;LOAD DATA LIGHTS
003054  016777  176006  175740          MOV     ADDR,@RKDA      ;LOAD DRIVE ADDRESS
003062  012777  000003  175724          MOV     #3,@RKCS        ;LOAD FUNCTION REGISTER
003070  105777  175720          TSTB    @RKCS           ;TEST FOR DONE
003074  100375                  BPL     .-4             ;WAIT FOR DONE
003076  005777  175712          TST     @RKCS           ;TEST FOR ERROR
003102  100014                  BPL     RDLINE          ;GO READ DATA NO CONTROL ERROR
```

```
003174  000000                          HALT                    ;LIGHTS CONTAIN DATA TO BE WRITTEN
003176  017700  175702                  MOV     @RKCS,R0        ;MOVE FUNCTION REGISTER TO DATA LIGHTS
003212  000000                          HALT                    ;DATA LIGHTS CONTAIN FUNCTION REGISTER
003214  017700  175704                  MOV     @RKER,R0        ;MOVE ERROR REGISTER TO DATA LIGHTS
003220  000000                          HALT                    ;DATA LIGHTS CONTAIN ERROR REGISTER
003222  017700  175700                  MOV     @RKDS,R0        ;MOVE DRIVE STATUS TO THE DATA LIGHTS
003226  000000                          HALT                    ;DATA LIGHTS CONTAIN DRIVE STATUS
003230  016700  175730                  MOV     DATA,R0         ;RESET THE DATA IN REGISTER ZERO
                                                                ;RETURN TO THE DISK DATA TEST TO TROUBLE
                                                                ;SHOOT THE PROBLEM
                                        ;
                                        ;
                                        ;RK11 HANDLER TO READ DATA AFTER IT IS WRITTEN
                                        ;ON SURFACE ZERO AND ONE
                                        ;
                                        ;
003234  012777  177777  175654  RDLINE: MOV     #-1,@RKWC       ;LOAD THE WORD COUNT
003242  012777  004304  175650          MOV     #OUTBUF,@RKBA   ;LOAD THE CURRENT ADDRESS
003250  016777  175712  175644          MOV     ADDR,@RKDA      ;LOAD THE DRIVE ADDRESS
003256  012777  000005  175630          MOV     #5,@RKCS        ;LOAD THE FUNCTION REGISTER
003264  105777  175624                  TSTB    @RKCS           ;TEST FOR DONE
003270  100375                          BPL     .-4             ;WAIT FOR DONE
003272  005777  175616                  TST     @RKCS           ;TEST FOR ERROR
003276  100014                          BPL     DTLINE          ;BRANCH IF NO CONTROL ERROR
003300  000000                          HALT                    ;DATA LIGHTS CONTAIN DATA TO BE READ
003302  017700  175606                  MOV     @RKCS,R0        ;MOVE CONTROL REGISTER TO DATA LIGHTS
003306  000000                          HALT                    ;DATA LIGHTS CONTAIN FUNCTION REGISTER
003310  017700  175610                  MOV     @RKER,R0        ;MOVE ERROR TO THE DATA LIGHTS
003314  000000                          HALT                    ;DATA LIGHTS CONTAIN ERROR REGISTER
003316  017700  175604                  MOV     @RKDS,R0        ;MOVE DRIVE STATUS TO THE DATA LIGHTS
003322  000000                          HALT                    ;DATA LIGHTS CONTAIN THE DRIVE STATUS
                                                                ;AND ERROR OCCURRED THE OPERATOR SHOULD
                                                                ;RETURN TO THE DISK DATA TEST TO
                                                                ;TROUBLE SHOOT HIS PROBLEM
003324  016700  175634                  MOV     DATA,R0         ;RESET DATA IN REGISTER ZERO
                                        ;
                                        ;
                                        ;
003330  020067  001050          DTLINE: CMP     R0,OUTBUF       ;WAS DATA READ CORRECTLY
003334  001404                          BEQ     LINEX           ;EXIT DATA CORRECT
003336  000000                          HALT                    ;DATA LIGHTS CONTAIN CORRECT DATA
003340  016700  001040                  MOV     OUTBUF,R0       ;MOVE DATA READ TO THE DATA LIGHTS
003344  000000                          HALT                    ;DATA LIGHTS CONTAIN DATA READ
                                                                ;USER SHOULD RETURN TO DISK DATA TO
                                                                ;TROUBLE SHOOT HIS PROBLEM
003346  000207          LINEX:  RTS     R7                      ;EXIT
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;RANDOM DATA GENERATOR SUBROUTINE
003350  016767  175574  175600  GENRAN: MOV     SAVE,WORK       ;#OF RANDOM WORDS
003356  012701  004304                  MOV     #OUTBUF,R1      ;START OF BUFFER
```

```
003262  016700  000120  RANDOM: MOV     LONUM,R0        ;SET UP R0 WITH 5 DIGITS LOW
003266  016704  000116          MOV     HINUM,R4        ;SET UP R1 WITH 5 DIGITS HIGH
003272  012703  000007          MOV     #7,R3           ;SET UP SHIFT COUNT
003276  005002                  CLR     R2              ;CLEAR R2
003300  006300          SHIFT:  ASL     R0              ;SHIFT R0 LEFT AND
003302  006104                  ROL     R4              ;ROTATE CARRY INTO LSB OF R1 INTO
```

```
003304  006102                          ROL     %2              ;ROTATE CARRY UT OF R1 INTO R2
003306  005303                          DEC     %3              ;DECREMENT R3
003310  001373                          BNE     SHIFT           ;CONTINUE SHIFT LOOP
003312  066700  000070                  ADD     LONUM,%0        ;ADDN IN NUMBER TO MAKE X 129
003316  005504                          ADC     %4              ;PROPOGATE CARRY
003320  066704  000064                  ADD     HINUM,%4        ;ADDN IN NUMBER TO MAKE X 129
003324  005502                          ADC     %2              ;PROPOGATE CARRY
003326  062700  001057                  ADD     #1057,%0        ;ADDN LOW CONSTANT
003332  005504                          ADC     %4              ;PROPOGATE CARRIES
003334  005502                          ADC     %2              ;PROPOGATE AGAIN
003336  062704  047401                  ADD     #47401,%4       ;ADDN HIGH CONSTANT
003342  005502                          ADC     %2              ;PROPOGATE CARRY
003344  062702  000006                  ADD     #6,%2           ;ADDN HIGHEST CONSTANT
003350  060200                          ADD     %2,%0           ;REPRIME R0 WITH HIGH DIGIT
003352  005504                          ADC     %4              ;PROPOGATE CARRY
003354  010067  000026                  MOV     %0,LONUM        ;PUT R0 BACK IN LONUM
003360  010021                          MOV     %0,(1)+         ;HOLD LONUM FOR PROGRAM
003362  005367  175470                  DEC     WORK
003366  001406                          BEQ     EXGEN
003370  010467  000014                  MOV     %4,HINUM        ;PUT R1 BACK IN HINUM
003374  010421                          MOV     %4,(1)+         ;HOLD HINUM FOR PROGRAM
003376  005367  175454                  DEC     WORK
003402  001327                          BNE     RANDOM
003404  000205                  EXGEN:  RTS     %5              ;RETURN TO PROGRAM
003406  000000                  LONUM:  0
003410  000000                  HINUM:  0
                                ;
                                ;
                                ;
                                ;THIS ROUTINE DETERMINES HOW MUCH MEMORY
                                ;IS ON THE SYSTEM, SO IT CAN DEFINE
                                ;THE LARGEST TRANSFER LENGTH,
                                ;
                                ;
003412  012777  000340  175362  EXTMEM: MOV     #340,@CSR       ;LOCK UP CPU
003420  012767  003666  174356          MOV     #MAXREF,4       ;SET UP I/O BUS TRAP
003426  012767  000340  174352          MOV     #340,6
003434  005067  175410                  CLR     SAVE
003440  005777  175404          REPMEM: TST     @SAVE           ;REFERENCE MEMORY
003444  022767  160000  175376          CMP     #160000,SAVE    ;TEST FOR 28K
003452  001001                          BNE     .+4             ;BRANCH IS LESS THAN 28K
003454  000404                          BR      MAXREF          ;
003456  062767  020000  175364          ADD     #20000,SAVE     ;SET UP NEXT MEMORY REFERENCE
003464  000765                          BR      REPMEM          ;REFERENCE MEMORY
                                ;
                                ;I/O BUS TRAP RETURNS YOU HERE
                                ;
                                ;THEN SET UP THE MAX, NUMBER OF WORDS THAT
                                ;CAN BE TRANSFERRED
003466  012767  000006  174310  MAXREF: MOV     #6,4            ;RESTORE I/O BUS TRAP
003474  005067  174306                  CLR     6
003500  005767  174336                  TST     42
003504  001410                          BEQ     RKMEM=6
```

```
003506  022767  002102  174326          CMP     #LOGICAL,42
003514  001404                          BEQ     RKMEM+6
003516  162767  003000  175324          SUB     #3000,SAVE
003524  000403                          BR      RKMEM
003526  162767  000500  175314          SUB     #500,SAVE       ;SUB, NON-EXISTENT MEMORY BANK
003534  162767  004304  175306  RKMEM:  SUB     #OUTBUF,SAVE    ;SUB, MEMORY USED BY PROGRAM
003542  000241                          CLC
003544  006067  175300                  ROR     SAVE            ;CONVERT FROM BYTES TO WORDS
003550  042767  000001  175272          BIC     #BIT0,SAVE      ;FUDGE EVEN NUMBER OF WORDS
003556  012706  001000                  MOV     #1000,%6        ;RESET STACK
003562  000205                          RTS     %5              ;RETURN.
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;ROUTINE TO DECODE EMT CALLS
                                ;EMT+1=TYPE ONE LINE OF TEXT
                                ;EMT+0=TYPE A SERIES OF LINES
003564  011600                  EMT%RP: MOV     (6),%0
003566  022740  104001                  CMP     #EMT+1,-(0)     ;WAS THE CALL EMT+1
003572  001050                          BNE     TYPS            ;NO!  TYPE A SERIES OF LINES OF TEXT
                                ;THIS IS THE SUB ROUTINE TO OUTPUT ASCII MESSAGES
                                ;
                                ;
                                ;
003574  011600                  TYP:    MOV     @%6,%0          ;GET THE ADDRESS THAT CONTAINS MESSAGE ADDRESS
003576  062716  000002                  ADD     #2,@%6          ;SET UP THE EXIT
003602  011000                          MOV     @%0,%0          ;ADDRESS OF MESSAGE IN REG,0
003604  112067  000102          TYPA:   MOVB    (0)+,TYPDAT     ;GET THE CHARACTER
003610  122767  000100  000074          CMPB    #100,TYPDAT     ;CHECK FOR @ SIGN
003616  001001                          BNE     TYPC            ;BRANCH IF NOT @ SIGN
003620  000002                          RTI                     ;CHARACTER WAS @  TERMINATOR EXIT
003622  122767  000045  000062  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR % SIGN  # CR
003630  001415                          BEQ     TYPF            ;BRANCH IF NOT % SIGN
003632  122767  000042  000052          CMPB    #42,TYPDAT      ;NOT % CHECK FOR # SIGN
003640  004767  000002                  JSR     %7,TYPD         ;PRINT CHARACTER IN TYPDAT
003644  000757                          BR      TYPA            ;FETCH NEXT CHARACTER
003646  116777  000040  175130  TYPD:   MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
003654  105777  175130          TYPE:   TSTB    @TPS            ;WAIT FOR FLAG
003660  100375                          BPL     .-4
003662  000207                  TYEXIT: RTS     %7              ;EXIT THIS ROUTINE
003664  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TTY
003672  004767  177750                  JSR     %7,TYPD
003676  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LINE FEED TO TTY
003704  004767  177736                  JSR     %7,TYPD         ;GO TYPE
003710  000735                          BR      TYPA
003712  000000                  TYPDAT: 0
                                ;THIS ROUTINE IS TO TYPE A SERIES OF MESSAGES
                                ;
                                ;
003714  011600                  TYPS:   MOV     @%6,%0          ;GET FIRST MESSAGE
003716  062716  000002                  ADD     #2,@%6          ;UPDATE ADDRESS FOR THE NEXT MESSAGE
003722  011067  000014                  MOV     @%0,TYPSR       ;ADDRESS OF NEXT MESAGE
003726  022767  177777  000006          CMP     #-1,TYPSR
```

```
003734  001001                          BNE     TYPSA
003736  000002                          RTI
003740  104001                  TYPSA:  EMT     +1
003742  000000                  TYPSB:  0
003744  000763                          BR      TYPS
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;POWER DOWN SEQUENCE FOR POWER FAIL
003746  012767  003772  174050 DOWN:   MOV     #UP,24          ;SET VECTOR TO POWER UP ADDRESS
003754  012777  000001  175032         MOV     #BIT0,@RKCS     ;CLEAR CONTROL
003762  105777  175026                 TSTB    @RKCS           ;TEST FOR READY
003766  100375                         BPL     .-4
003770  000000                         WAIT                    ;WAIT FOR POWER UP
                                ;
                                ;
                                ;
                                ;POWER UP SEQUENCE FOR POWER FAIL
                                ;
                                ;
003772  012767  003746  174024 UP:     MOV     #DOWN,24        ;SET VECTOR UP FOR POWER DOWN
004000  104001                         EMT     +1              ;REPORT POWER FAIL OCCURRED
004002  004220                         PFMES
004004  000167  175260                 JMP     RESTART         ;GO RESTART
                                ;
                                ;
                                ;
```

```
                                ;
                                ;
                                ;OCTAL TO ASCII CONVERT ROUTINE
                                ;
                                ;ENTER ROUTINE AS FOLLOWS
                                ;JSR%5,CONV
                                ;ADDR#=ADDRESS OF NUMBER TO BE CONVERTED
                                ;ADDR BYTE=LSB OF WHERE ASCII IS GOING
                                ;ASCII#=THE NUMBER OF ASCII CHAR, TO BE CONVERTED
                                ;
                                ;
004010  013567  000054        CONV:   MOV     @(5)+,ACNVX     ;VALUE OF # TO BE CONVERTED
004014  012501                        MOV     (5)+,%1         ;ASCII ADDR
004016  012502                        MOV     (5)+,%2         ;# OF ASCII CHAR
004020  060201                        ADD     %2,%1
004022  016703  000042        ACVN:   MOV     ACNVX,%3
004026  042703  177770                BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT OCTAL#
004032  062703  000060                ADD     #60,%3          ;SET UP ASCII#
004036  110341                        MOVB    %3,-(1)         ;STORE ASCII CHAR
004040  042767  000007  000022        BIC     #7,ACNVX
004046  006067  000016                ROR     ACNVX           ;ROTATE OCTAL#
004052  006067  000012                ROR     ACNVX
004056  006067  000006                ROR     ACNVX
004062  005302                        DEC     %2              ;-1 FROM ASCII CHAR COUNT
004064  001356                        BNE     ACVN
004066  000205                        RTS     %5              ;EXIT # CONVERTED
004070  000000        ACNVX:  0                               ;WORK REGISTER
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
```

```
004072  000000                  TEXBUF:0
                                ;
                                ;
                                ;
004074      040                 CON1:   .ASCII  /  DRIVE(S) ON LINE:/
004075      040
004076      104
004077      122
004100      111
004101      126
004102      105
004103      050
004104      123
004105      051
004106      040
004107      117
004110      116
004111      040
004112      114
004113      111
004114      116
004115      105
004116      072

                                ;
                                ;
                                ;ERROR MESSAGE HEADERS
                                ;

        004120                          .EVEN
004120      045                 MED2:   .ASCII  /%ERROR #/
004121      105
004122      122
004123      122
004124      117
004125      122
004126      040
004127      043

                                ;
                                ;
004130      045                 MED3A:  .ASCII  /%:/
004131      072

                                ;
                                ;
                                ;MESSAGE TRAILERS
                                ;
                                ;
004132      040                 MES1:   .ASCII  /       RK:A #/
004133      040
004134      040
004135      040
004136      040
```

```
004137      040
004140      040
004141      122
004142      113
004143      104
004144      101
004145      040
004146      100

                                ;
                                ;
                                ;
004147      040                 MES1A:  .ASCII  /       RKER @/
004150      040
004151      040
004152      040
004153      040
004154      040
004155      040
004156      122
004157      113
004160      105
004161      122
004162      040
004163      100

                                ;
                                ;
                                ;
004164      040                 MES2:   .ASCII  /       RKCS @/
004165      040
004166      040
004167      040
004170      040
004171      040
004172      040
004173      122
004174      113
004175      103
004176      123
004177      040
004200      100

                                ;
004201      040                 MES2A:  .ASCII  /       RKDS @/
004202      040
004203      040
004204      040
004205      040
004206      040
004207      040
004210      122
004211      113
004212      104
004213      123
```

```
004214      040
004215      100

                                    ;
                                    ;
                                    ;
004216      045                     MES8I   ,ASCII  /X@/
004217      100

                                    ;
                                    ;
004220      045                     PFMES1  ,ASCII  /XPOWER FAIL HAS OCCURRED@/
004221      120
004222      117
004223      127
004224      105
004225      122
004226      040
004227      106
004230      101
004231      111
004232      114
004233      040
004234      110
004235      101
004236      123
004237      040
004240      117
004241      103
004242      103
004243      125
004244      122
004245      122
004246      105
004247      104
004250      100

                                    ;
                                    ;
004251      045                     END1    ,ASCII  /XEND@/
004252      105
004253      116
004254      104
004255      100

                                    ;
                                    ;
                                    ;
004256      045                     NODRV1  ,ASCII  /XDRIVE ZERO OFF LINE@/
004257      104
004260      122
004261      111
004262      126
004263      105
004264      040
```

```
004265      132
004266      105
004267      122
004270      117
004271      040
004272      117
004273      126
004274      126
004275      040
004276      114
004277      111
004300      116
004301      105
004302      120
```

```
004304                          .EVEN
                                ;
004304  000000          OUTBUF: 0
        000001                  .END
```

ACNVX	004070	LETSGO	001430	TKB	001006
ACVN	004022	LETSWR	001364	TKS	001012
ADDR	001066	LINECK	003026	TPB	001004
ADREXI	002752	LINEX	003246	TPS	001010
ADROK	002434	LOADBA	003016	TYEXIT	003662
ADTST	002370	LODEN	002520	TYP	003574
BASAV	001040	LOGICA	002102	TYPA	003604
BIT0	000001	LONUM	003406	TYPC	003622
BIT1	000002	MAXREF	003466	TYPD	003646
BIT10	002000	MEMEND	001060	TYPDAT	003712
BIT11	004000	MES1	004132	TYPF	003664
BIT12	010000	MES1A	004147	TYPG	003676
BIT13	020000	MES2	004164	TYPS	003714
BIT14	040000	MES2A	004201	TYPSA	003740
BIT15	100000	MES8	004216	TYPSB	003742
BIT2	000004	NEWFUN	001062	UP	003772
BIT3	000010	NEWSK	001474	VECTOR	001030
BIT4	000020	NODRV	004256	WATSEK	001704
BIT5	000040	NOP	000240	WCSAV	001036
BIT6	000100	OUTBUF	004304	WORK	001056
BIT7	000200	PASS	001042	WORK1	001054
BIT8	000400	PFMES	004220	WRCHEC	000507
BIT9	001000	PRIORI	001034	WRITE	000503
BLKCNT	002634	RANDOM	003262		
BLOCK	001052	RDLINE	003134		
BLSZ	002500	REFMEM	003440		
CHKCYL	002716	REPDV	001242		
CHKDAT	001774	REPEND	002112		
CHKSUR	002670	RESTAR	001270		
CON1	004074	RESTOR	001304		
CONM	001170	RKBA	001020		
CONV	004010	RKCS	001014		
COUNT	001044	RKDA	001022		
CSR	001032	RKDONE	002140		
CYTST	002416	RKDS	001026		
DATA	001064	RKER	001024		
DOWN	003746	RKMEM	003534		
DRIVE	001046	RKNODV	001224		
DTLINE	003230	RKWC	001016		
DTMOD	001374	RORBLK	002562		
EMTRP	003564	SAVE	001050		
END	004251	SECOVE	001752		
ERROR	002170	SECSEE	001692		
EXGEN	003404	SEEK	000111		
EXIT	002362	SETBLK	002526		
EXTMEM	003412	SHIFT	003300		
GENRAN	003250	SKOVER	001542		
HED2	004120	SOFTER	002344		
HED3A	004130	SR	001000		
HINUM	003410	START	001070		
INTHN	002122	STATUS	001032		
LETSEE	001442	TEXBUF	004072		

ERRORS DETECTED: 0

RUN-TIME: 7 SECONDS

5K CORE USED

```
DRIVER+7     Open[1]
      +10    Transfer
      +11    Close[1]
      +12    Special Functions[1]
```

The problem, of course, is the start address. There is always the obvious solution, that of assembling the driver at a fixed location so that each program using it can immediately reference the location chosen. This, however, ceases to be convenient when the program itself has to avoid the area given to the driver. A more general method is to relocate the driver as dictated by the program using it, thus taking advantage of the position-independent nature of the driver. The Absolute Loader, described in the Paper Tape Software Handbook (DEC-11-GGPB-D), Chapter 6, provides the capability of continuing a load from the point at which it ended. Using this facility to enter the driver immediately after the program, the program itself might contain the following code to call the subroutine to perform the transfer illustrated in section 3.1.3:

```
        MOV    PC,R1              ;GET DRIVER START ADDRESS
        ADD    #PRGEND-.,R1
        MOV    PC,RØ              ;GET TABLE ADDRESS
        ADD    #TABLE+12-.,RØ     ;& SET UP AS SHOWN
         .                        ;...IN SECTION 3.1.3
         .
         .
        CMP    -(RØ),-(RØ)        ;FINAL POINTER ADJUSTMENT
        MOV    RØ,@R1             ;STORE IN DRIVER LINK
        CLR    -(SP)              ;GET BYTE SHOWING...
        MOVB   1Ø(R1),@SP         ;...TRANSFER OFFSET
        ADD    (SP)+,R1           ;COMPUTE ADDRESS
        JSR    PC,@R1             ;GO TO DRIVER
         .
         .
         .
PRGEND:
       .END
```

This technique may be extended to cover situations in which several drivers are used by the same program, provided that it takes account of the size of each driver (this being already known because of prior assembly) and that the drivers themselves are always loaded in the same order.

For example, to access the second driver, the above sequence would be modified to:

[1]If the routine is not provided, these are Ø.

```
        MOV    PC,R1            ;GET DRIVER 1 ADDRESS
        ADD    #PRGEND-.,R1
        ADD    #DVR1SZ,R1       ;STEP TO DRIVER 2
         .
         .
         .
DVR1SZ=
PRGEND:
       .END
```

An alternative method may be to use the Relocatable Assembler PAL-11S in association with the Linker program LINK-11S, both of which are available through the DECUS Library. The start address of each driver is identified as a global. Any calling program need, therefore, merely include a corresponding .GLOBL statement, e.g., .GLOBL DT.

3.3 Device-independent Usage

As mentioned earlier, the drivers are designed for use in a device-independent environment, i.e., one in which a calling program need not know in advance which driver has been associated with a table for a particular execution run. One application of this type might be to allow line-printer output to be diverted to some other output medium because the line-printer itself is currently not available. Another might be to provide a general program to analyze data samples although these on one occasion might come directly from an Analog to Digital converter and on another be stored on a DECtape, because the sampling rate was too high to allow immediate evaluation.

As a rule, programs of this type should be written to cater for all the facilities that any one device might offer, but not necessarily all of them. For instance, the program should ask for start-up procedures because it may sometime use a paper tape punch which provides them, even though it may normally use DECtape which does not. As noted in section 2.1.1, the driver table contains an indication of its capabilities to cater for this situation. The program can thus examine the appropriate item before calling the driver to perform some action. As an example, the code to request start-up procedures might be (assuming RØ already set to List Address):

```
MOV    #DVRADD,R1       ;GET DRIVER ADDRESS
TSTB    2(R1)           ;BIT 7 SHOWS...
BPL    NOOPEN           ;...OPEN ROUTINE PRESENT
MOV    RØ,@R1           ;STORE TABLE ADDRESS
CLRB   -(SP)            ;BUILD ADDRESS
MOVB   7(R1),@SP        ;...OF THIS ROUTINE
ADD    (SP)+,R1
```

```
        JSR   PC,@R1            ;...& GO TO IT
                                ;FOLLOWED POSSIBLY BY
                                ;WAIT AND COMPLETION
                                ;PROCESSING
NOOPEN:                         ;RETURN TO COMMON OPERATION
```

Similarly, the indicators show whether the device is capable of performing input or output or both, whether it can handle ASCII data or Binary data, whether it is a bulk storage device capable of supporting a directory structure or is a terminal-type device requiring special treatment and so on. Other table entries show the device name as identification and how many words it might normally expect to transfer at a time (in 16-word units). All of the information may readily be examined by the calling program, thus enabling the use perhaps of a common call sequence for any I/O operation, as for example:

```
        MOV   #DVRADR, R5       ;SET DRIVER START
        JSR   R5, IOSUB         ;CALL SET UP SUB
        BR    WAIT              ;SKIP TABLE FOLLOWING ON RETURN
        .WORD 1Ø                ;TRANSFER REQUIRED
        .WORD 1Ø3               ;BLOCK NO.
        .WORD BUFFER            ;BUFFER ADDRESS
        .WORD -256.             ;WORD COUNT
        .WORD 4Ø4               ;READ FROM UNIT 1
        .WORD RETURN            ;EXIT ON COMPLETION
        .WORD Ø                 ;RESERVED
WAIT:                           ;CONTINUE HERE...
        .                       ;WHILE TRANSFER IN PROGRESS
        .
        .
        .

IOSUB:  MOV   @SP,RØ            ;PICK UP DRIVER ADDR
        MOV   R5,R1             ;SET POINTER TO LIST
        TST   (R1)+             ;BUMP TO COLLECT CONTENT
        .                       ;ROUTINE CHECKS ON DEVICE..
        .                       ;..CAPABILITY USING R1
        .                       ;..TO ACCESS LIST &
        .                       ;..RØ THE DRIVER TABLE
        .                       ;IF O.K...
        MOV   @R1,R1            ;GET ROUTINE OFFSET
        ADD   RØ,R1
        CLR   -(SP)             ;USE IT TO BUILD
        MOVB  @R1,@SP           ;...ENTRY POINT
        ADD   RØ,@SP
        JSR   PC,@(SP)+         ;CALL DRIVER
        RTS   R5                ;EXIT TO CALLER
```

The calling program, or a subroutine of the type just illustrated, may also wish to take advantage of a further feature mentioned earlier: the fact that when a driver is already occupied its first word must be

non-zero. The driver itself does not clear this word except in special cases shown in the description for the driver concerned. If the program itself always ensures that it is set to zero between driver tasks, this word forms a suitable Driver-busy flag. Under DOS, in fact, the program parameter list is extended to allow additional words to provide linkage between lists as a queue of which the list indicated in the driver first word is the first link.

The preceding paragraphs are intended merely to indicate possible ways of incorporating the drivers available into the type of environment for which they were designed. The user will probably find others. However, he should read carefully the more detailed description of the driver structure in Appendix A and the individual driver specifications before determining the final form of his program.

In particular, one general word of warning is appropriate here. Although most drivers normally set up an operation and then wait for an interrupt to produce a completion state, there are some cases in which the driver can finish its required task without an interrupt, e.g., "opening" a paper-tape reader involves only a check on its status. Moreover, where "Special Functions" are concerned, the driver routine may determine from the code indicated that the function is not applicable in its case and will, therefore, have nothing to do. In those cases, the driver clears the intermediate return address from the processor stack and takes the completion return immediately. Special problems may arise, however, if the driver concerned may be covering several tasks, any of which may cause a queue for the driver's services under DOS. To overcome these problems, the driver expects to be able to refer to flags outside the scope of the list described so far. This may mean that a program using such a driver may also need to extend the list range to cover this possibility. Extreme care will then be needed.

APPENDIX A

I-O DRIVERS WITHIN THE DISK OPERATING SYSTEM

The principal function of an I/O driver is to satisfy the requirement of a Monitor processing routine for the transfer of a block of data in a standard format to or from the device it represents. This will involve both setting up the device hardware registers to cause the transfer and its control under the interrupt scheme of PDP-11, making due allowance for peculiar device characteristics (e.g., conversion to or from ASCII if some special code is used).

It may also include routines for handling device start-up or shut-down such as punching leader or trailer, and for making available to the user certain special features of the device, such as rewind of magtape.

A.1 Driver Structure

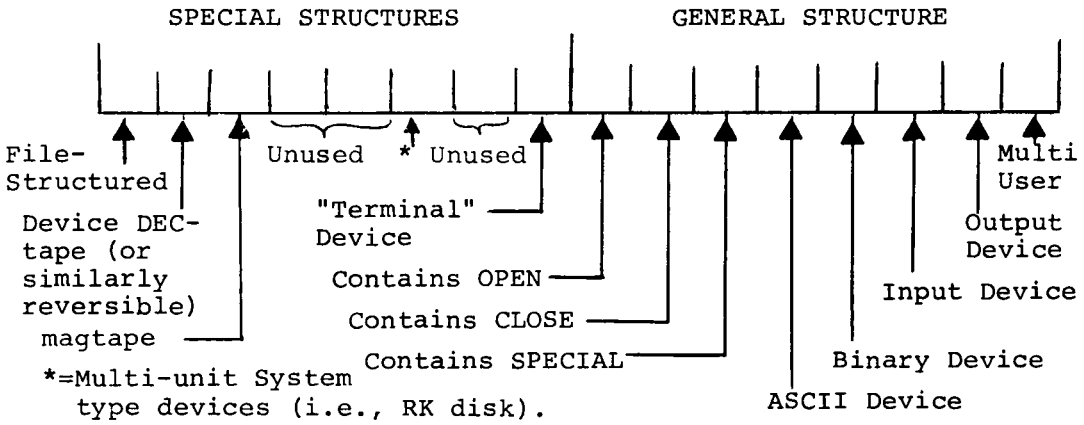
In order to provide a common interface to the monitor, all drivers must begin with a table of identifying information as follows:

DVR:

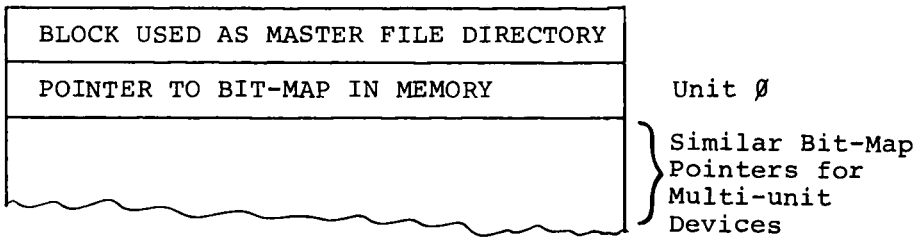
BUSY FLAG (initially Ø)	
FACILITY INDICATOR (expanded below)	
Offset to Interrupt Routine*	Standard Buffer Size in 16-word Units.
Offset to OPEN Routine *	Priority for Interrupt Service
Offset to CLOSE Routine *	Offset to Transfer Routine *
Space	Offset to Special Functions*
DEV	NAME (Packed Radix-5Ø)

Offsets marked * will enable calling routine to indicate routine required. They will be considered as an unsigned value to be added to the start address of the driver. This may mean that with a 256 maximum, the instruction referenced by the offset will be JMP or BR (routine).

Bits in the Facility Indicator Word define the device for monitor reference:



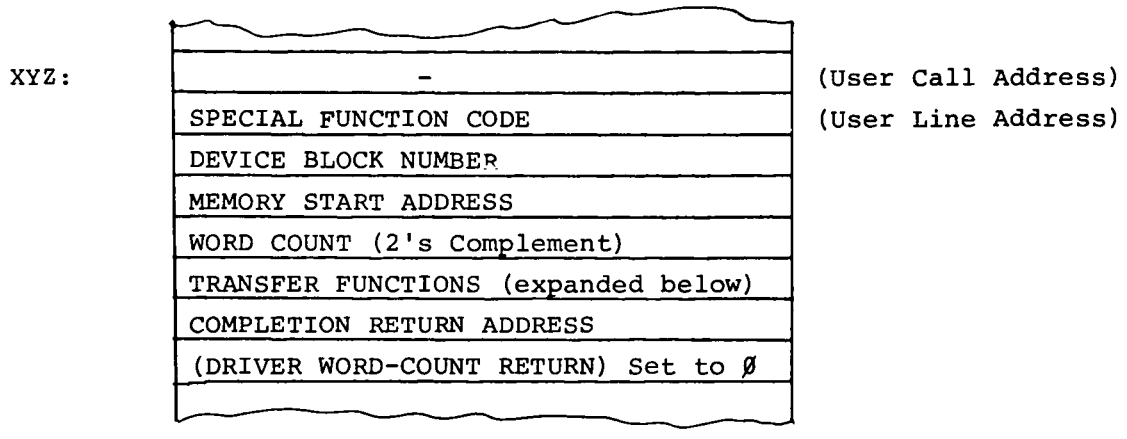
The table should be extended as follows if the device is file-structured:



The driver routines to set up the transfer and control it under interrupt, and possibly for OPEN, CLOSE, and SPECIAL, follow the table. Their detailed operation will be described later.

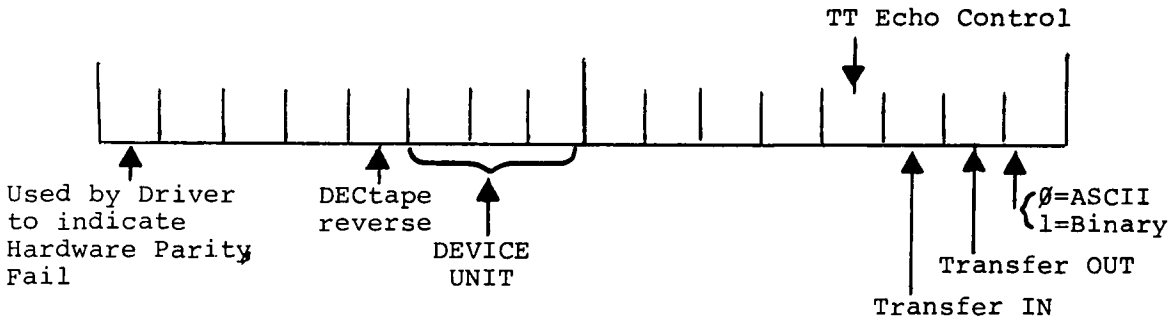
A.2 Monitor Calling

When a Monitor I/O processing routine needs to call the driver, it first sets up the parameters for the driver operation in relevant words of the appropriate DDB[1], as follows:



[1]Dataset Data Block - in full, a 16-word table which provides the main source of communication between the Monitor drivers and a particular set of data being processed on behalf of a using program.

The relevant content of the Transfer Function word is as follows:



Provided that the Facility Indicator in the Driver Table described above shows that the driver is capable of satisfying the request, both from the point of view of direction and mode and of the service required, the Monitor routine places in Register 1 the relative byte address of the entry in the Driver Table containing the offset to the routine to be used (e.g., for the Transfer routine, this would be 1Ø). It then calls the Driver Queue Manager, using JSR PC,S.CDB.

The Driver Queue Manager ensures that the driver is free to accept the request, by reference to the Busy Flag (Word Ø of the driver table). If this contains Ø, the Queue Manager inserts the address of the DDB from Register Ø and jumps to the start of the routine in the driver using Register 1 content to evaluate the address required. If the driver is already occupied, the new request is placed in a queue linking the appropriate DDB's for datasets waiting for the driver's services. It is taken from the queue when the driver completes its current task. (This is done by a recall to the Queue Manager from the routine just serviced, using JSR PC,S.CDQ.)

On entry to the Driver Routine, therefore, the address following the Monitor routine call remains as the "top" element of the processor stack. It can be used by the driver in order to make an immediate return to the Monitor (having initiated the function requested), using RTS PC. It should also be noted that the Monitor routine will have saved register contents if it needs them after the device action. The driver may thus freely use the registers for its own operations.

When the driver has completely satisfied the Monitor request, it should return control to the Monitor using the address set into the DDB. On such return, Register Ø must be set to contain the address of the DDB just serviced and since the return will normally follow hardware interrupt, Registers Ø-5 at the interrupt must be stored on top of the stack.

A.3 Driver Routines

A.3.1 TRANSFER

The sole purpose of the TRANSFER routine is to set the device in motion. As indicated above, the information needed to load the hardware registers is available in the DDB, whose address is contained in the first word of the driver. Conversion of the stored values is, of course, the function of the routine. It must also enable the interrupt; however, it need not take any action to set the interrupt vectors as these will have been preset by the Monitor when the driver is brought into core. Having then given the device GO, an immediate return to the calling processor should be made by RTS PC.

A.3.2 Interrupt Servicing

The form of this routine depends upon the nature of the device. In most drivers it will fall into two parts, one for handling the termination of a normal transfer and the other to deal with reported error conditions.

For devices which are word or byte-oriented, the routine must provide for individual word or byte transfers, with appropriate treatment of certain characters (e.g., TAB or Null) and for their conversion between ASCII or binary and any special device coding scheme, until either the word count in the DDB is satisfied or an error prevents this. On these devices, the most likely cause for such error is the detection of the end of the physical medium; its treatment will vary according to whether the device is providing input or accepting output. The calling program will usually need to take action in the former case and the driver should merely indicate the error by returning the unexpired portion of the word count in DDB Word 7 on exit to the Monitor. Output End of Data, however, will, in general, require operator action. To obtain this, the driver should call the Error Diagnostic Print routine within the Monitor by:

```
MOV    DEVNAM,-(SP)          ;SHOW DEVICE NAME
MOV    #4Ø2,0(SP)            ;SHOW DEVICE NOT READY
IOT                          ;CALL E.D.P.
```

On the assumption that the operator will reset the device for further output and request continuation, the driver must follow the above sequence with a Branch or Jump to produce the desired resumption of the transfer.

Normal transfer handling on blocked devices (or those like RF11 Disk which are treated as such) is probably simpler since the hardware takes care of individual words or bytes and the interrupt only occurs on completion. Errors may arise from many more causes, and their handling is, as a result, much more complex and device dependent. In general, those which indicate definite hardware malfunctions must lead to the situation in which the operator must be informed by diagnostic message and the only recourse after rectification will be to start the program over.

At the other end of the scale there are errors which the driver itself can attempt to overcome by restarting the transfer - device parity failure on input is a common example. If a retrial, or several, still does not enable a satisfactory conclusion, the driver should normally allow programmed recovery and merely indicate the error by Bit 17 of DDB word 5. Nevertheless, because the program may wish to process the data despite the error, the driver should attempt to transfer the whole block requested if this has not already been effected. Between these two extremes, the remaining forms of error must be processed according to the type of recovery deemed desirable.

Whether the routine uses processor registers for its operation or not will naturally depend on considerations of the core space saved against the time taken to save the user's content. However, on completion (or error return) to the Monitor, as indicated in an earlier paragraph, the calling routine expects the top of the stack to contain the contents of all Registers Ø-5 and Register Ø to be set to the address of the DDB just serviced. The driver must, therefore, provide for this.

A.3.3 OPEN

This routine need be provided only for those devices for which some hardware initialization is required by the user. It should not

normally appear in drivers for devices used in a file-oriented manner. Its presence must be indicated by the appropriate bit (Bit 7) in the driver table Facility Indicator.

The routine itself may vary according to the transfer direction of the device. For output devices, the probable action required is the transmission of appropriate data, e.g., CR/LF at a keyboard terminal, form-feed at a printer, or null characters as punched leader code, and for this a return interrupt is expected. The OPEN routine should then be somewhat similar to that for TRANSFER in that it merely sets the device going and makes an interim return via RTS PC, waiting until completion of the whole transmission before taking the final return address in the DDB.

On the other hand, an input OPEN will likely consist of just a check on the readiness of the device to provide data when requested. In this case, the desired function can be effected without any interrupt wait. The routine should, therefore, take the completion return immediately. Nevertheless, it must ensure that the saved PC value on top of the stack from the call to S.CDB is appropriately removed before exit. In the case of drivers which can only service one dataset at a time (i.e., Bit Ø of their Facility Pattern word is set to Ø) and can never, therefore, be queued, it will be sufficient merely to use TST (SP)+ to effect this. A multi-user driver, however, must allow for the possibility that it may be recalled to perform some new task already waiting in a queue. This is shown by the byte at DDB-3 being non-Ø. In this case, the intermediate return to the routine originally requesting the new task has already been made directly by S.CDB. The address now on top of the stack is the return to the routine, whose task the driver has just completed and which has called S.CDQ to dequeue the driver. This return must be taken when the first routine has performed its Completion Return processing. Moreover, this first routine expects to exit as from an interrupt. When a driver is recalled from a queue, it must simulate this interrupt. A possible sequence might be:

```
        MOV    DRIVER, RØ        ;PICK UP DDB ADDRESS
        MOV    (SP)+,R5          ;SAVE INTERIM RETURN
        TSTB   -3(RØ)            ;COME FROM QUEUE?
        BEQ    EXIT
        MOV    @#177776,-(SP)    ;IF SO, STORE STATUS
        MOV    R5,-(SP)          ;...& RETURN
        SUB    #14,SP            ;DUMMY SAVE REGS
EXIT:   JMP    @14(RØ)
```

A.3.4 CLOSE

As with OPEN, this routine should provide for the possibility of some form of hardware shut down such as the punching of trailer code and is not necessary for file-structured devices. Moreover, it is likely to be a requirement for output devices only. If it is provided, Driver Table Facility Indicator (Bit 6) must be set.

Again, the probable form is initialization of the hardware action required, with immediate return via RTS PC and eventual completion return via the DDB-stored address.

A.3.5 SPECIAL

This routine may be included if either the device itself contains the hardware to perform some special function or there is a need for software simulation of such hardware on other devices, e.g., tape rewind. It should not be provided otherwise. Its presence must be indicated by Bit 5 of the Facility Indicator.

The function itself is stored by the Monitor as a code in the DDB as shown earlier. When called, the driver routine must determine whether such function is appropriate in its case. If not, the completion return should be taken immediately with prior stack clearance, as discussed under OPEN. For a recognized function, the necessary routine must be provided. Again, its exit method will depend upon the necessity for an interrupt wait or otherwise.

A.4 Drivers for Terminals

The rate of input from terminal devices is normally dictated externally by the operator, rather than being program-driven; moreover, for both input and output, the amount of data to be transferred on each occasion may be a varying value, i.e., a line rather than a block of standard size. Furthermore, there may be problems with the conflict between echo of input during output. As a result, drivers for such devices will demand special treatment.

Normal output operation, i.e., .WRITE by the program, is handled by the Monitor Processor. On recognizing that the device being used is a terminal, as shown by Bit 8 of the facility indicator, this routine always causes a driver transfer at the end of the user line, even though the internal buffer has not been filled. The driver, however, is given the whole of a standard buffer, padded as necessary with

nulls. Provided the driver can ignore these, the effect is that of just a line of output.

Input control on the other hand, must remain driver responsibility. Overcoming the rate problem will, in most cases, require circular buffering within the driver until demanded by the Monitor. At this point, transfer of data already in should occur. If this is sufficient to fill the monitor buffer, the driver can await the next request before further transfer onward. If insufficient, it should operate as any other device and use subsequent interrupts to continue to satisfy the Monitor request. It must, nevertheless, stop any transfer at the end of a line in normal operation. In order to allow the Monitor to continue, the driver must simulate the filling of the buffer by null padding (of no consequence, since terminals are by nature character-based). (Normal operation, of course, means response to user .READ's and is indicated by the size of the buffer to be filled, namely the driver standard. Should the user be requesting .TRAN's, the buffer size will vary from the standard in all likelihood and the driver may then assume he requires operation as a normal device -- complete buffer fill-up before return.)

Where input echo is a further complexity, there will doubtless be other requirements. If the echo is made immediately after the input, it may be desirable to have a second buffer to cater for the likely situation that the echo will not exactly match its origin. On the other hand, if the echo is held for any length of time, perhaps to provide correct relations between program-driven output and the echo, the second buffer could be too expensive. A larger input buffer and routines to allow for several outputs to one input character while sitting on that character might be more convenient. The conflict between such echo and program-driven output will require controlled switching within the driver input and output handlers.

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 & PDP-12
Digital Software News for the PDP-11
Digital Software News for the PDP-9/15 Family

These newsletters contain information applicable to software available from Digital's Program Library, Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning Digital's Software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problem to:

Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

These forms which are provided in the software kit should be fully filled out and accompanied by teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software and manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest Digital Field office or representative. U.S.A. customers may order directly from the Program Library in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

DECUS
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

PDP-11 Device Driver Package
DEC-11-NIZB-D
February 1972.

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability.

Did you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Please state your position. _____ Date: _____

Name: _____ Organization: _____

Street: _____ Department: _____

City: _____ State: _____ Zip or Country _____

— — — — — — — — — — — — — — — — Fold Here — — — — — — — — — — — — — — — — — —

— — — — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

DEC-11-NIZB-D

P D P - 1 1

D E V I C E D R I V E R P A C K A G E

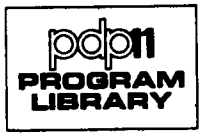
FEBRUARY 1972

COPYRIGHT © 1971, 1972 BY DIGITAL EQUIPMENT CORPORATION

SOFTWARE SUPPORT CATEGORY

The software described in this document is supported by DEC under Category I, as defined on page ii of this document.

THIS DOCUMENT IS FOR INFORMATION PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.



SOFTWARE SUPPORT CATEGORIES

Digital Equipment Corporation (DEC) makes available four categories of software. These categories reflect the types of support a customer may expect from DEC for a specified software product. DEC reserves the right to change the category of a software product at any time. The four categories are as follows:

CATEGORY I
Software Products Supported at no Charge

This classification includes current versions of monitors, programming languages, and support programs provided by DEC. DEC will provide installation (when applicable), advisory, and remedial support at no charge. These services are limited to original purchasers of DEC computer systems who have the requisite DEC equipment and software products.

At the option of DEC, a software product may be recategorized from Category I to Category II for a particular customer if the software product has been modified by the customer or a third party.

CATEGORY II
Software Products that Receive Support for a Fee

This category includes prior versions of Category I programs and all other programs available from DEC for which support is given. Programming assistance (additional support), as available, will be provided on these DEC programs and non-DEC programs when used in conjunction with these DEC programs and equipment supplied by DEC.

CATEGORY III
Pre-Release Software

DEC may elect to release certain software products to customers in order to facilitate final testing and/or customer familiarization. In this event, DEC will limit the use of such pre-release software to internal, non-competitive applications. Category III software is only supported by DEC where this support is consistent with evaluation of the software product. While DEC will be grateful for the reporting of any criticism and suggestions pertaining to a pre-release, there exists no commitment to respond to these reports.

CATEGORY IV
Non-Supported Software

This category includes all programs for which no support is given

CONTENTS

		Page
CHAPTER 1.	INTRODUCTION	1-1
CHAPTER 2.	DRIVER FORMAT	2-1
2.1	Structure	2-1
2.1.1	Driver Interface Table	2-1
2.1.2	Set-up Routines	2-1
2.1.3	Interrupt Servicing	2-1
2.1.4	Error Handling	2-2
2.2	Interface to the Driver	2-2
2.2.1	Control Interface	2-2
2.2.2	Interrupt Interface	2-2
CHAPTER 3.	STAND-ALONE USAGE	3-1
3.1	Driver Assembled with Program	3-1
3.1.1	Setting Interrupt Vector	3-1
3.1.2	Parameter Table for Driver Call	3-1
3.1.3	Calling the Driver	3-2
3.1.4	User Registers	3-3
3.1.5	Return from Driver	3-4
3.1.6	Irrecoverable Errors	3-5
3.1.7	General Comment	3-6
3.2	Drivers Assembled Separately	3-6
3.3	Device-independent Usage	3-8
APPENDIX A	I-O DRIVERS WITHIN THE DISK OPERATING SYSTEM	A-1
A.1	Driver Structure	A-1
A.2	Monitor Calling	A-2
A.3	Driver Routines	A-4
A.3.1	TRANSFER	A-4
A.3.2	Interrupt Servicing	A-4
A.3.3	OPEN	A-5
A.3.4	CLOSE	A-7
A.3.5	SPECIAL	A-7
A.4	Drivers for Terminals	A-7

CHAPTER 1

INTRODUCTION

Subroutines to handle I/O transfers between a PDP-11 and each of its peripheral devices are developed as required for use within the Disk Operating System (DOS). These subroutines are made available within an I/O Utilities Package for the benefit of PDP-11 users who have configurations unable to support DOS or who wish to run programs outside DOS control.

All the subroutines associated with one peripheral device together form an entity which is known as a Driver. The purpose of this document is to provide a general description of a driver and to show how it may be used in a stand-alone environment. The unique properties of each driver are discussed in separate documents issued as supplements to this one. The I/O Utilities Package for any system is determined by the peripherals of that system. Thus the full documentation for a particular package consists of this document and the applicable supplements.

Within this document, Chapter 2 consists of an outline of the established driver structure and its interface to the program using it. Chapter 3 then illustrates how a stand-alone program can match this interface in order to make immediate use of each driver as supplied within the package. For the benefit of those users who require a more detailed description of the driver format, perhaps so that they can write their own drivers for other unsupported devices in a similar fashion, the standard specification for DOS driver has been attached as Appendix A. It is assumed that the reader is familiar with the basic hardware concepts of the PDP-11 as described in the PDP-11 Processor Handbook and with the Paper Tape Software as described in the Programming Handbook (DEC-11-GGPC-D).

%%%%%%

CHAPTER 2

DRIVER FORMAT

2.1 Structure

The basic principle of all drivers under the DOS Monitor is that they must present a common interface to the routines using them in order to provide for device-independent operation. The subroutines are structured to meet this end. Moreover, the driver may be loaded anywhere in memory under Monitor control. Its code must always, therefore, be position-independent.

The detailed description of a driver is found in Appendix A. This chapter is concerned with driver interfaces.

2.1.1 Driver Interface Table

The first section of each driver consists of a table which contains, in a standard format, information on the nature and capabilities of the device it represents and entry pointers to each of its subroutines. The calling program may then use this table as required, regardless of the device being called.

2.1.2 Setup Routines

Each driver is expected to handle its device under the PDP-11 interrupt system. When called by a program, therefore, a driver subroutine merely initiates the action required by setting the device hardware registers appropriately. It then returns to the calling program by a standard subroutine exit.

The main setup routine prepares for a data transfer to or from the device, using parameters supplied by the calling program. Normally, blocks of data will be moved at each transfer. The driver will only return control to the program when the whole block has been actioned or when it is unable to continue because there is no more data available.

The driver may also contain subroutines by which the calling program may request start-up or shut-down action, such as leader or trailer code at a paper-tape punch, or some special function provided by the device hardware (or a software simulation of that for some similar device), e.g., "rewind" of a magnetic tape (or DECtape).

2.1.3 Interrupt Servicing

The nature of the driver routine to service device interrupts is

particularly dependent upon the extent of the hardware provisions of the device for controlling transfers. In general, the driver determines the cause of the interrupt and checks whether the last action was performed correctly or was prevented by some error condition. If more device action is needed to satisfy the program, the driver again initiates that action and takes a normal interrupt exit. If the program request has been fully met, control is returned to the program at an address supplied at the time of the call.

2.1.4 Error Handling

Device errors may be handled in two ways. There are some errors for which recovery can be programmed; the driver will, if appropriate, attempt this itself (as in the case of parity or timing failure on a bulk-storage device) or will recall the program with the error condition flagged (as at the end of a physical paper tape). Other errors will normally require action externally, perhaps by an operator. For the latter, the driver calls a common error handler based on location 34 (IOT call) with supporting information on the processor stack.

2.2 Interface to the Driver

2.2.1 Control Interface

The principal link between a calling program and any driver subroutine is the first word of the driver table. In order to provide the control parameters for a device operation, the calling program prepares a list in a standardized form and places a pointer to the list in the driver link. The called driver then uses the pointer to access the parameters. If the driver need then return status information, it may again place this in the list area via the link-word.

The first word of the driver also may act as an indicator in that while it remains Ø, the driver is not already busy upon some task, whereas when the word contains a list-pointer, the driver is assumed to be busy. Since most drivers can only support one job at a time, the link-word state can be significant.

2.2.2 Interrupt Interface

Although the driver will always expect to use the interrupt system, it does not itself ensure that its interrupt vector in the memory area below 400 has been set up correctly; the Monitor under DOS takes care of this. However, the Driver Table contains the necessary information to allow the vector to be set correctly.

CHAPTER 3

STAND-ALONE USAGE

Because each driver is designed for operation within the device-independent framework of DOS Monitor, it may be similarly used in other applications. Possible methods will be discussed later. However, since the easiest way to use the driver is to assemble it with the program requiring it, this will be described first.

3.1 Driver Assembled with Program

3.1.1 Setting Interrupt Vector

As noted in Section 2.2.2, the calling program must first correctly set the device transfer vector within memory locations 0-377. The address of the driver's interrupt entry point can be identified on the source listing by the symbolic name which appears as the content of the Driver Table Byte, DRIVER+5. The priority level at which the driver expects to process the interrupt is at byte DRIVER+6. For a program which can use position-dependent code, the setup sequence may be:

```
MOV   #DVRINT, VECTOR      ;SET INT. ADDRESS
MOVB  DRIVER+6, VECTOR+2   ;SET PRIORITY
CLRB  VECTOR+3             ;CLEAR UPPER STATUS BYTE
```

(where the Driver Table shows at DRIVER +5: .BYTE DVRINT-DRIVER).

If the program must be position-independent, it may take advantage of the fact that the Interrupt Entry address is actually stored as an offset from the start of the driver, as illustrated above. In this case, a sample sequence might be:

```
MOV   PC,R1                ;GET DRIVER START
ADD   #DRIVER-.,R1
MOV   #VECTOR,R2           ;...& VECTOR ADDRESSED
CLR   @R2                  ;SET INT. ADDRESS
MOVB  5(R1),@R2            ;...AS START ADDRESS+OFFSET
ADD   R1,(R2)+
CLR   @R2                  ;SET PRIORITY
MOVB  6(R1),@R2
```

3.1.2 Parameter Table for Driver Call

For any call to the driver, the program must provide the list of

control arguments mentioned in Section 2.2.1. This list must adhere in general to the following format:[1]

```
[SPECIAL FUNCTION CODE][2]
[BLOCK NO.][3]
STARTING MEMORY ADDRESS FOR TRANSFER
NO. OF WORDS to be transferred (2's complement)
STATUS CONTROL showing in Bits:
  Ø-2:  Function (octally 2=WRITE, 4=READ)[4]
  8-1Ø: Unit (if Device can consist of several, e.g., DECtape)
  11:   Direction for DECtape travel (Ø = Forward)
ADDRESS for RETURN ON COMPLETION
[RESERVED FOR DRIVER USE][5]
```

The list itself may be assembled into the required format if its content will not vary. The driver may return information in the area as described in a later paragraph; however, this will not corrupt the program data and it is removed by the driver before it begins its next operation.

On the other hand, most programs will probably wish to use the same area for the lists for several tasks or even between different drivers. In this case, the program must contain the necessary routine to set up the list for each task before making the driver call, perhaps as illustrated in the next paragraph. It must be noted, however, that the driver may wish to refer to the list again when it is recalled by an interrupt or to return information to the calling program. Therefore, the list must not be changed until any driver has completed a function requested; for concurrent operations, different list areas must be provided.

3.1.3 Calling the Driver

To enable the driver to access the parameter list, the program must set the first word of the driver to an address six bytes less than that

---

[1] In some cases, it may be further extended as discussed in later paragraphs.

[2] Required only if Driver is being called for Special Function.

[3] Required only if the Device is bulk storage (e.g., Disk or DECtape).

[4] Most devices transfer words regardless of their content, i.e. ASCII or Binary. Some devices, e.g., Card Reader, may be handled differently for the two modes; for these, Bit Ø must also be set to indicate ASCII=Ø, Binary=1. (In these cases, the driver always produces or accepts ASCII even though the device itself uses some other code.)

[5] This word may be omitted if the device is bulk storage (see below).

of the word containing MEMORY START ADDRESS. It may then call the driver subroutine required directly by a normal JSR PC,xxxx call.

As an example, the following position-independent code might appear in a program which wishes to read Blocks #1ØØ-1Ø3 backward from DECtape Unit into a buffer starting at address BUFFER:

```
        MOV    PC,RØ                    ;GET TABLE ADDRESS
        ADD    #TABLE+12-.,RØ
        MOV    PC,@RØ                   ;GET & STORE...
        ADD    #RETURN-.,@RØ            ;...RETURN ADDRESS
        MOV    #54Ø4,-(RØ)              ;SET READ REV. UNIT 3
        MOV    #-1Ø24.,-(RØ)            ;4 BLOCKS REQUIRED
        MOV    PC,-(RØ)                 ;GET & STORE
        ADD    #BUFFER-.,@RØ            ;...BUFFER ADDRESS
        MOV    #1Ø3,-(RØ)               ;START BLOCK
        CMP    -(RØ),-(RØ)              ;SUBTRACT 4 FROM POINTER
        MOV    RØ,DT                    ;SET DRIVER LINK
        JSR    PC,DT.TFR                ;GOTO TRANSFER ROUTINE
WAIT:   .                               ;RETURNS HERE WHEN
        :                               ;...TRANSFER UNDERWAY
RETURN: .                               ;RETURNS HERE WHEN
        :                               ;... TRANSFER COMPLETE
TABLE:  .WORD Ø                         ;LIST AREA SET
        .WORD Ø                         ;...BY ABOVE SEQUENCE
        .WORD Ø
        .WORD Ø
        .WORD Ø
```

3.1.4 User Registers

During its setup operations for the function requested, the driver assumes that Processor Registers Ø-5 are freely available for its purpose. If their contents are of value, the program must save them before the driver is called.

While servicing intermediate interrupts, the driver may need to save or restore these registers. It expects to have available two subroutines for the purpose (provided by the Monitor under DOS) It accesses them via addresses in memory locations 44 (SAVE) and 46 (RESTORE) using the sequence:

```
MOV    @#44,-(SP)          ;OR 'MOV    @#46,-(SP)
JSR    R5,@(SP)+
```

The program must, therefore, contain these subroutines. They might, for example, be as follows:

```
SAVE:       MOV   R4,-(SP)      ;SAVE RØ-4
            MOV   R3,-(SP)      ;...R5 SAVED BY CALL
            MOV   R2,-(SP)
            MOV   R1,-(SP)
            MOV   RØ,-(SP)
            MOV   R5,PC         ;EXIT TO CALLER

RESTOR:     INC   (SP)+         ;FORGET CALL R5
            MOV   (SP)+,RØ      ;RESTORE RØ-4
            MOV   (SP)+,R1
            MOV   (SP)+,R2
            MOV   (SP)+,R3
            MOV   (SP)+,R4
            RTS   R5            ;R5 RESET ON EXIT
```

It must also ensure that their start addresses are set into the correct locations.

At its final interrupt, the driver always saves the contents of Registers Ø-5 before returning control to the calling program completion return.

3.1.5 Returns from Driver

As shown in the example in section 3.1.3, the driver returns control to the calling program immediately after the JSR as soon as it has set the device in motion. The program may then wait or carry out some alternative operations until the driver signals completion by returning at the address supplied, i.e., RETURN above. Prior to this, the program should not attempt to access the data being read in, or to refill a buffer being written out.

The program routine beginning at address RETURN will vary according to the device in use. In general, the driver has given control to the routine for one of two reasons, namely, the function has been satisfactorily performed, or it cannot be carried out due to some hardware failure with which the driver is unable to cope, though the program may. If the latter, the driver uses the STATUS word in the program list to show the cause:

Bit 15 = 1	indicates that a device parity or timing failure has occurred and the driver has not been able to overcome this, perhaps after several attempts.
Bit 14 = 1	shows that the end of the data available has been reached.

The driver places in RØ the content of its first word as a pointer to the list concerned.

In addition, the driver may have transferred only some of the data required. In this case, it will show, in the RESERVED word of the program list, a negative count of the words not transferred in addition to setting Bit 14 of the STATUS. As mentioned in the note in Section 3.1.2, this applies only to non-bulk storage devices. The drivers for DECtape or Disks[1] always endeavor to complete the full transfer, even beyond a parity failure, or they take more drastic action (see Section 3.1.6).

It is thus the responsibility of the program RETURN routine to check the information supplied by the driver in order to verify that the transfer was satisfactory and to handle the error situations accordingly.

In addition, the routine must contain a sequence to take care of the Processor Stack, Registers, etc. As noted earlier, the driver takes the completion return address after an interrupt and has saved Registers 0-5 on the stack above the Interrupt Return Address and Status. The program routine should, therefore, contain some sequence to restore the processor to its state prior to such interrupt, e.g., using the same Restore subroutine illustrated earlier:

```
MOV    @#46,-(SP)          ;CALL REGISTER RESTORE
JSR    R5,@(SP)+
  .
  .
  .
RTI                        ;RETURN TO INTERRUPTED PROG.
```

3.1.6 Irrecoverable Errors

All hardware errors other than those noted in the previous paragraph are more serious in that they cannot normally be overcome by the program or the driver on its behalf. Some of these could be due to an operator fault, such as an omission to turn a paper tape reader on or to set the correct unit number on a DECtape transport. Once the operator has rectified the problem, the program could continue. Other errors, however, will require hardware repair or even software repair, e.g., if the program asks for Block 2000 on a device having a maximum of 1000. In general, all these errors will result in the driver placing identifying information on the processor stack and calling IOT to produce a trap through location 34.

[1]This includes RF11 Disk: although this is basically word-oriented, it is assumed to be subdivided into 64-word blocks.

Under DOS, the Monitor provides a routine which prints a teleprinter message when this occurs. In a stand-alone environment, the program using the driver must itself contain the routine to handle the trap (unless the user wishes to modify the driver error exits before assembly). The handler format will depend upon the program. Should it wish to take advantage of the information supplied by the driver, the format is as follows:

(SP):	Return Address }	Stored by IOT Call
2(SP):	Return Status }	
4(SP):	Error No. Code	generally unique to driver
5(SP):	Error Type Code:	1 = Recoverable after Operator Action 3 = No recovery
6(SP):	Additional Information	such as content of Driver, Control Register, Driver Identity, etc.

As a rule, the driver will expect a return following the IOT call in the case of errors in Type 1 but will contain no provision following a return from Type 3.

3.1.7 General Comment

The source language of each driver has been written for use with the DOS version of the Assembler which requires certain statements which will not be accepted by the Paper Tape Software PAL-11A, in particular: .TITLE & .GLOBL. These should be edited out before the source is used. Similarly, an entry in the driver table gives the device name as .RAD5Ø 'DT' to obtain a specially packed format used internally by DOS. If the user still wishes to keep the name, for instance for identification purposes as discussed in section 3.3, .RAD5Ø might easily be changed to .ASCII without detrimental effect, or it can be replaced with .WORD Ø .

3.2 Drivers Assembled Separately

Rather than assemble the driver with every program requiring its availability, the user may wish to hold it in binary form and attach it to the program only when loaded. This is readily possible; the only requirement is that the start address of the driver should be known or can be determined by the program.

The example in section 3.1.2 showed that the Interrupt Servicing routine can be accessed through an offset stored in the Driver Table. The same technique can be used to call the setup subroutines, as these also have corresponding offsets in the Table, as follows: